

# MODELO DE REFERENCIA PARA REPRESENTACIÓN GRÁFICA DE OBJETOS INTELIGENTES EN REALIDAD AUMENTADA PARA LA WEB DE LAS COSAS



Monografía de Trabajo de Grado

**YURI VIVIANA DAZA GAVIRIA  
JUAN FELIPE VALENCIA MOSQUERA**

*Universidad del Cauca*  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Sistemas  
Popayán  
2018

# MODELO DE REFERENCIA PARA REPRESENTACIÓN GRÁFICA DE OBJETOS INTELIGENTES EN REALIDAD AUMENTADA PARA LA WOT



**YURI VIVIANA DAZA GAVIRIA  
JUAN FELIPE VALENCIA MOSQUERA**

**TRABAJO DE GRADO**

Presentado como requisito para optar al título de Ingeniero De Sistemas

Director

**PhD. Miguel Ángel Niño Zambrano**

*Universidad del Cauca*

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas

Popayán, Febrero 2018

**NOTA DE ACEPTACIÓN**

---

---

---

---

**PRESIDENTE DEL JURADO**

---

**JURADO**

**Popayán, 2018**



## TABLA DE CONTENIDO

CAPÍTULO 1	PRESENTACIÓN DEL PROYECTO.....	1
1.1.	CONTEXTO GENERAL.....	1
1.2.	DECLARACIÓN DEL PROBLEMA.....	2
1.3.	ESCENARIO DE MOTIVACIÓN.....	3
1.4.	OBJETIVOS.....	4
1.4.1.	Objetivo general.....	4
1.4.2.	Objetivos específicos.....	4
1.5.	ALCANCE DEL PROYECTO Y CONTRIBUCIONES.....	4
1.5.1.	Alcance del proyecto.....	4
1.5.2.	Contribuciones.....	5
1.6.	ESTRUCTURA DE LA MONOGRAFÍA.....	6
CAPÍTULO 2	MARCO TEÓRICO.....	9
2.1.	INTRODUCCIÓN.....	9
2.2.	CONCEPTOS FUNDAMENTALES.....	9
2.2.1.	Internet de las Cosas (Internet of Things - IoT).....	9
2.2.2.	Web de las Cosas (Web of Things - WoT).....	10
2.2.3.	Interacción Semántica.....	12
2.2.4.	Arquitectura del Proyecto.....	12
2.2.5.	Objetos Inteligentes.....	13
2.2.6.	Contexto y Contexto-Consciencia.....	13
2.2.7.	Representaciones Gráficas en 2D.....	15
2.2.8.	Representaciones Gráficas en 3D.....	15
2.2.9.	Realidad Aumentada.....	15
2.2.10.	Herramientas de Aplicación (Términos).....	16
2.3.	TRABAJOS PREVIOS.....	18
CAPÍTULO 3	MODELO DE REFERENCIA PARA REPRESENTACIÓN GRAFICA DE IO.....	21
3.1.	DEFINICIÓN DEL CONJUNTO GENERAL DE LINEAMIENTOS PARA LA CREACIÓN DE MODELOS GRÁFICOS DE OBJETOS INTELIGENTES EN REALIDAD AUMENTADA.....	21
3.2.	DEFINICIÓN FORMAL DEL MODELO DE REFERENCIA.....	24
3.2.1.	Fase 1: Ámbito del modelo de referencia.....	24
3.2.2.	Fase 2: Definición formal de ARMoWoT.....	25
3.2.3.	Fase 3: Diseño de prueba preliminar del modelo de referencia.....	40



CAPÍTULO 4	CONSTRUCCIÓN DEL ESTUDIO DE CASO.....	52
4.1.	IMPLEMENTACIÓN DEL MODELO DE REFERENCIA ARMoWoT .....	52
4.1.1.	Paso 1: Análisis y diseño del escenario .....	52
4.1.2.	Paso 2: Investigar herramienta adecuada .....	52
4.1.3.	Paso 3: Examinar el repositorio de Plugins creados.....	53
4.1.4.	Paso 4: Revisar formato guía de lineamientos generales para representaciones graficas de IO.....	53
4.1.5.	Paso 5: Retroalimentar formato guía de lineamientos para representaciones graficas de IO.....	53
4.1.6.	Paso 6: Establecer propiedades del Plugin .....	54
4.1.7.	Paso 7: Seleccionar metodología de desarrollo .....	55
4.1.8.	Paso 8: Creación técnica del Plugin.....	56
4.1.9.	Paso 9: Testing del Plugin.....	57
4.1.10.	Paso 10: Actualizar el repositorio con los nuevos Plugins creados.....	57
4.1.11.	Paso 11: Usar Plugin en el nuevo proyecto AR .....	57
4.2.	CONSTRUCCIÓN DEL PROYECTO AR PARA LA WOT .....	58
4.2.1.	Análisis .....	58
4.2.2.	Diseño.....	62
4.2.3.	Construcción .....	63
4.2.4.	Prototipo 1.....	64
4.2.5.	Prototipo 2.....	69
CAPÍTULO 5	EVALUACIÓN Y ANÁLISIS.....	71
5.1.	FASE DE DEFINICIÓN DE LOS INDICADORES.....	71
5.1.1.	Caracterización del personal a evaluar .....	71
5.2.	FASE DE IMPLEMENTACIÓN DE INDICADORES.....	72
5.2.1.	Diagrama de Gantt para el Plan de pruebas .....	72
5.3.	FASE DE ANÁLISIS Y RETROALIMENTACIÓN AL MODELO.....	72
5.3.1.	Presentación de los resultados .....	72
5.3.2.	Retroalimentación al modelo ARMoWoT.....	78
CAPÍTULO 6	CONCLUSIONES Y TRABAJO FUTURO.....	79
6.1.	CONCLUSIONES .....	79
6.1.1.	CONCLUSIONES EN CUANTO A PRODUCTOS OBTENIDOS .....	79
6.1.2.	CONCLUSIONES EN CUANTO A LA INVESTIGACIÓN .....	80
6.1.3.	CUMPLIMIENTO DE OBJETIVOS.....	79
6.2.	TRABAJO FUTURO.....	81

Modelo de Referencia para Representación Gráfica de Objetos  
Inteligentes en Realidad Aumentada Para la WoT.

---





## LISTA DE ILUSTRACIONES

Ilustración 1 - Etapas de la metodología propuesta para el desarrollo del proyecto .....	5
Ilustración 2 - Aplicaciones de la IoT.....	10
Ilustración 3 - Conectando cosas a la Web .....	11
Ilustración 4 - Arquitectura de interacción semantica entre objetos inteligentes de la IoT. 12	
Ilustración 5 - Ciclo de vida del contexto .....	14
Ilustración 6 - Virtuality Continuum (VC) .....	16
Ilustración 7 - Formato Guía de Lineamientos Generales para crear representaciones gráficas de los IO de acuerdo a las características de sensores, actuadores y servicios que los componen. ....	22
Ilustración 8 - Ejemplo instanciado del Formato Guía de Lineamientos Generales para los IO de un Termo Inteligentes.....	23
Ilustración 9 - Diagrama de Flujo para el Modelo de Referencia ARMoWoT .....	27
Ilustración 10 – Esquema de sensores, actuadores y servicios que componen un objeto inteligente. ....	30
Ilustración 11 - Test de Plugin creado.....	35
Ilustración 12 - Zona iterable para la creación del Plugin .....	38
Ilustración 13 - Ejemplo de Comunicación en Unity 3D.....	45
Ilustración 14 - Plugin 2D Temperatura.....	46
Ilustración 15 - Plugin 2D Temperatura y su ventana de personalización .....	47
Ilustración 16 - Plugin 3D Temperatura para Termo Inteligente .....	47
Ilustración 17 - Plugin 3D Temperatura y su ventana de personalización para Termo Inteligente.....	48
Ilustración 18 - Escena inicial de Termo Inteligente .....	50
Ilustración 19 - Representación gráfica en AR para el IO Termo Inteligente .....	50
Ilustración 20- Termo Inteligente con diferentes temperaturas.....	51
Ilustración 21 - Capa de interacción.....	59
Ilustración 22 - Escenario de Domótica.....	60
Ilustración 23 - Prototipo 1 .....	65
Ilustración 24 – Jerarquía de carpetas para los Plugins en el entorno de Unity 3D .....	66
Ilustración 25 - Representación 2D para sensores del escenario de Domótica .....	67
Ilustración 26 - Representación 3D para sensores del escenario de Domótica.....	67
Ilustración 27 - Representación 3D para actuadores del escenario de Domótica (Representación opcional de actuadores).....	68
Ilustración 29 - Prototipo 2 .....	69
Ilustración 30 - Escenario 2, pantalla principal .....	70
Ilustración 31 - Diagrama de Gantt para el Plan de Pruebas.....	72
Ilustración 32 - Resultados de la Escala de Likert.....	78



## LISTA DE TABLAS

Tabla 1 - Tabla de Características de Términos.....	17
Tabla 2 - Variación de la terminología del producto dependiente de la tecnología seleccionada para el desarrollo software .....	18
Tabla 3 - Metadatos de los Plugins.....	33
Tabla 4 - Lista de chequeo para test del Plugin creado.....	37
Tabla 5 - Metadatos de los Plugins pertenecientes a la prueba preliminar del modelo de referencia ARMoWoT. ....	44
Tabla 6 - Metadatos de los Plugins pertenecientes al escenario de Domótica del modelo de referencia ARMoWoT .....	55
Tabla 7 - Distribución de Pruebas para los participantes .....	73
Tabla 8 - Consolidado de la Evaluación .....	77





## CAPÍTULO 1 PRESENTACIÓN DEL PROYECTO

### 1.1. CONTEXTO GENERAL.

La computación ubicua propone un cambio radical en la forma en que se comunican las personas entre sí y con su entorno, así podemos decir que la computación ubicua es la integración de la informática al entorno natural de las personas [1]. Con base en lo anterior, se generaron tecnologías como la Internet de las cosas (Internet of Things - IoT), la cual se centra en la interconexión de elementos que cotidianamente rodean a las personas [2, 3] y la Web de las cosas (Web of Things - WoT) que se centra en acercar el mundo físico al mundo digital en la Red Informática Mundial (World Wide Web - WWW) [4].

Estas tecnologías cada vez son más accesibles y económicas, permitiendo en los últimos años su producción en masa, además se ha visto la necesidad de generar sistemas capaces de interconectar las cosas y entender el contexto que lo rodea [5]. Sin embargo, existen inconvenientes por resolver, como lo es la falta de estandarización de hardware y software, la conectividad y la heterogeneidad, generando así problemas de escalabilidad y dificultad en el procesamiento de datos masivos (Big Data) [6].

Por lo tanto, Mainetti, V.Mighali y Patrono, establecieron arquitecturas enfocadas principalmente en la comunicación entre los objetos de la IoT [7]. Las investigaciones recientes, trabajan en la definición de diversos protocolos a nivel físico tales como: Bluetooth, IEEE 802.15.4 [8], el 6LoWPAN [9] y protocolos a nivel sintáctico como: el protocolo de aplicación restringida (The Constrained Application Protocol - COAP) <sup>1</sup>[10] protocolo de transporte de telemetría de mensajes Queue Server (Message Queue Telemetry Transport – MQTT) y el protocolo extensible de mensajería y comunicación de presencia (Extensible Messaging and Presence Protocol – XMPP). Los trabajos relacionados para unificar el acceso a los dispositivos proponen la creación de Middlewares, un ejemplo es la propuesta realizada por Mainetti, et al. [7], donde se presenta una arquitectura formada por distintos protocolos, con el objetivo de encontrar los dispositivos disponibles físicamente y extraer sus métodos de virtualización, para poder interactuar con ellos en “una interfaz común”. Así, el presente proyecto integra un middleware definido con el fin de enfocarse principalmente en la representación digital de los dispositivos [11], representando gráficamente todos los datos que arrojan los diferentes objetos inteligentes pertenecientes a la web de las cosas.

La interfaz común de los middlewares principalmente se ha enfocado en la presentación de la información de los metadatos y datos de los sensores a través de estándares como el lenguaje de marcado extensible (Extensible Markup Language – XML) y notación de objetos JavaScript (JavaScript Object Notation – JSON), sin tener en cuenta la representación gráfica, para el desarrollo de aplicaciones para la WoT.

Para establecer una interfaz común de representación gráfica de dispositivos de la WoT, no solamente se trata de obtener interfaces genéricas o estandarizadas, sino que

---

<sup>1</sup> COAP es un protocolo software a nivel de aplicación pensado para ser usado en dispositivos electrónicos simples permitiendo que puedas comunicarse sobre Internet.



incorporen conceptos como la transparencia, la facilidad de uso e interfaces intuitivas, además de que permita la gestión del contexto<sup>2</sup> [5] y la personalización. Las investigaciones que se han desarrollado al respecto como la de Guzmán-Álvarez [16] y la de A.Martínez-Ospina [39], han encontrado una solución en las técnicas de Realidad Aumentada (Augmented Reality - AR) [12]. La AR es una forma de mostrar el entorno que nos rodea visto a través de un dispositivo tecnológico, combinando los elementos reales con elementos virtuales, generando así un elemento de la realidad mixta en tiempo real [13].

Con respecto a los proyectos de AR se resalta el desarrollado por Leppanen, et al. [14], el cual usa agentes móviles que permite a los usuarios asignar interfaces<sup>3</sup> específicas a códigos de respuesta rápida (Quick Response – QR), mostrando así más información acerca de los objetos reales y de manera personalizada, sin embargo, el desarrollo de las interfaces está hechas a medida y por lo tanto es difícil de implementar en otros escenarios.

En el proyecto de Domingo [15], se muestra una visión general de la utilidad que tiene la IoT frente a los problemas de las personas con discapacidades, da a conocer algunos escenarios que pueden proveer solución a problemas visuales, auditivos y físicos. También introduce múltiples problemáticas que aún no tienen solución, tal como la personalización. La personalización en este proyecto se entiende como: la adecuación de la representación digital en AR de un recurso IoT de acuerdo a las preferencias de sus usuarios. Estas preferencias deben entenderse en la parte visual de un objeto prefabricado bidimensional o tridimensional, como son: tamaño, color y textura, entre otros.

A nivel local se tiene el proyecto de Guzmán-Álvarez [16] y Niño-Zambrano [11], los cuales proponen una arquitectura para la interacción semántica de objetos inteligentes en la IoT. Siendo cada uno de los objetos inteligentes representados por medio de una ontología, esta ontología es llamada objeto semántico. Los autores proponen una metodología para generar interfaces que utilicen la técnica de AR, basada en la arquitectura del Modelo Semántico, sin embargo, no se preocupa por la representación gráfica de los Objetos Inteligentes (Intelligent Object - IO).

## 1.2. DECLARACIÓN DEL PROBLEMA.

A través del contexto general, se percibe la ausencia de investigaciones relacionadas con la representación adecuada para los objetos inteligentes en el contexto de la WoT. Por lo tanto, se genera la importancia de contar con una infraestructura común de modelos gráficos personalizables, para la implementación de soluciones con AR en la WoT. Con el fin de tener una representación unificada de la información, minimizando los costos en el desarrollo y logrando su adaptación en otros escenarios. Así, se plantea la siguiente pregunta de investigación: **¿Cómo crear representaciones graficas de los objetos inteligentes en una interfaz AR para la WoT, que sean reutilizables y personalizables, de acuerdo a su contexto de aplicación?**

---

<sup>2</sup> El contexto se refiere al dominio de aplicación e intereses de los usuarios acerca de los servicios que puede proveer un objeto. Ejemplo: Utilizar un sensor de temperatura para medir la temperatura de una persona o utilizarlo para medir la temperatura de un cuarto en una casa, dónde se cambia tanto la entidad de interés: de persona a cuarto, como también el contexto: de salud a climático.  
<sup>3</sup> La interfaz es una de las formas en que se trasmite instrucciones que nos informan sobre su uso. La superficie de un objeto (real o virtual), nos habla por medio de sus formas, texturas, colores, etc.



El enfoque de solución propuesto en el presente proyecto, es la construcción de un modelo de referencia para la creación de modelos gráficos en realidad aumentada para la WoT, a través de objetos prefabricados<sup>4</sup> y desplegados mediante *Plugins*<sup>5</sup>, que se pueden personalizar de acuerdo a los sensores y servicios del recurso IoT representado. Aunque se tiene en cuenta que el término Plugin, puede cambiar de acuerdo a la tecnología seleccionada para el desarrollo del estudio de caso. La reutilización en esta solución se da al reusar código y otros componentes visuales, generando así objetos adaptables a los diversos contextos (ahorro en tiempo, dinero y optimización de la aplicación). Por último, para el manejo del contexto se hace uso de la arquitectura de interacción semántica propuesta por Niño-Zambrano [11].

### 1.3. ESCENARIO DE MOTIVACIÓN.

El diario vivir, viene marcado por diferentes características que permiten observar profundos cambios en el comportamiento del ser humano y en la manera de realizar sus actividades cotidianas a lo largo de la historia. Uno de esos factores que cambia de manera drástica los comportamientos entre las personas, culturas e incluso la naturaleza del ser humano es la tecnología. Debido al crecimiento tecnológico se pueden observar cambios en los sitios de trabajo, los hogares y en las personas, antes y después de aparecer los primeros aparatos electrónicos como: el radio, el televisor, los primeros ordenadores personales y actualmente los teléfonos inteligentes (Smartphone). A medida que ha pasado el tiempo han surgido nuevas tecnologías que aportan grandes beneficios a los seres humanos, dentro de ellas se destaca la AR, siendo una tecnología que permite expandir los sentidos y la percepción del entorno real, sin embargo, sus plataformas de desarrollo aún se encuentran en evolución. Actualmente la AR se está integrando constantemente a otras tecnologías y a dispositivos de fácil acceso como el Smartphone, esto hace que se abran muchas posibilidades de integración con la IoT, ya que esta tecnología permite que los objetos de la cotidianidad estén conectados a internet y otorguen información necesaria a las personas. Todas estas tecnologías se encuentran en evolución y ya se puede acceder a ellas con mayor facilidad. Hoy en día existe una gran competencia entre distintos desarrollos que involucran AR tanto a nivel de Hardware como Software, y es previsible que, a través de la aceptación del público, el software y sus campos de dominio tiendan a evolucionar, haciendo que la tecnología se masifique en su consumo.

En resumen, la presente investigación se motiva a generar la alianza de las tecnologías de realidad aumentada y la IoT, en búsqueda de otorgar información de los datos y metadatos de IO a usuarios, mediante representaciones gráficas en AR. Cabe resaltar que en la actualidad se perciben ausencia de investigaciones relacionadas con la debida creación de representaciones gráficas de los objetos inteligentes basados en modelo gráficos 2D<sup>6</sup> o 3D<sup>7</sup> en el contexto de la WoT. Por lo tanto, el reto es generar una infraestructura común de modelos gráficos personalizables y reutilizables, para la implementación de soluciones con AR en la WoT. Con el fin de tener una representación unificada de la información,

---

<sup>4</sup> Es un modelo gráfico al cual se le ha adicionado componentes (scripts o APIs) para definir sus comportamientos en una interfaz de AR. Por definición estos modelos son reutilizables en otras interfaces que utilicen igual entorno de desarrollo.

<sup>5</sup> Añade una funcionalidad adicional o una nueva característica al entorno de desarrollo, amplían así la funcionalidad de la plataforma

<sup>6</sup> Un modelo gráfico en 2D está compuesto por un espacio bidimensional.

<sup>7</sup> Un modelo gráfico en 3D está compuesto por un espacio tridimensional. Es una réplica matemática de un objeto, ya sea real o imaginario



minimizando costos en el desarrollo y buscando su adaptación en otros escenarios, representando de manera gráfica los IO y brindando soluciones capaces de obtener una visualización en AR de los sensores, actuadores y servicios que los componen. Haciendo uso de la arquitectura propuesta por Niño-Zambrano [11] y el escenario de interacción semántica definido por Riobamba y Guerrero [17], se realiza una exploración documental que permite determinar los elementos metodológicos y de diseño más adecuados para la representación gráfica de los IO mediante la técnica de AR, utilizando tecnología accesible y teniendo en cuenta lo que la comunidad científica recomienda usar, para desarrollos de cualquier representación gráfica en distintos escenarios de aplicación.

## 1.4. OBJETIVOS.

### 1.4.1. Objetivo general.

Proponer un modelo de referencia para la creación de modelos gráficos de objetos inteligentes [11] en realidad aumentada para la WoT, a través del desarrollo de *Plugins* personalizables y reutilizables en diferentes contextos de la WoT.

### 1.4.2. Objetivos específicos.

- Definir un conjunto de lineamientos para la creación de modelos gráficos de objetos inteligentes, de tal manera que permita:
  - i. Establecer la representación gráfica del IO de acuerdo a las características tales como: dispositivos, sensores, actuadores y servicios.
  - ii. Definir un mecanismo para su personalización y reutilización en interfaces AR, de acuerdo a su contexto de aplicación.
  - iii. Proponer la forma de crear *Plugins* para AR, que recojan las representaciones gráficas definidas del IO.
- Desarrollar una interfaz de AR para la WoT en estudios de caso que serán seleccionados en el transcurso del proyecto.
- Evaluar la capacidad de personalización y reutilización de los *Plugins* en otros contextos de aplicación.

## 1.5. ALCANCE DEL PROYECTO Y CONTRIBUCIONES.

### 1.5.1. Alcance del proyecto

Debido a que los productos que se van a generar en el modelo de referencia, usan diferentes metodologías para su desarrollo, se tomó la decisión de implementar un enfoque metodológico, el cual utiliza una metodología diferente para cada etapa de la investigación. Para el presente proyecto se tendrá en cuenta tres etapas principales (ver Ilustración 1): conceptualización, construcción y análisis - retroalimentación. Para cada etapa se elabora una serie de actividades que están compuestas por fases para el producto a obtener.



Adicionalmente, al final de cada etapa se genera una retroalimentación al modelo de referencia.

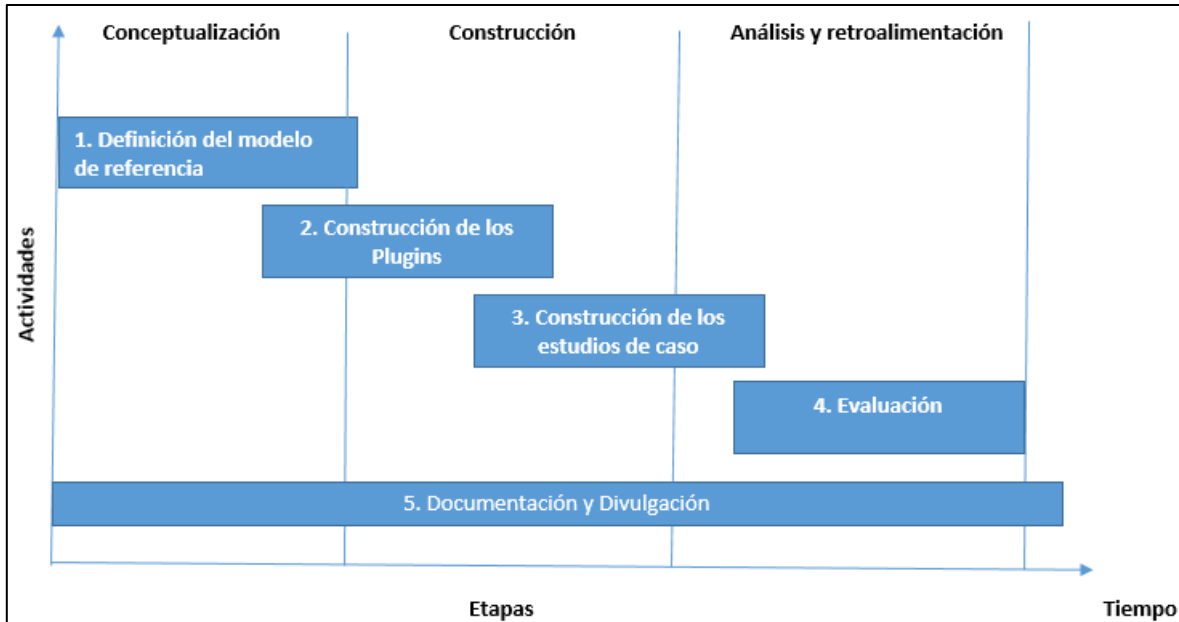


Ilustración 1 - Etapas de la metodología propuesta para el desarrollo del proyecto

Con lo anterior, el modelo de referencia propuesto se enfocará en tener un alcance distribuido entre sus objetivos. Siendo el primero el desarrollo del modelo de referencia el cual ocupara una parte considerable del proyecto (60%), por consiguiente, la construcción de los estudios de caso (25%) y su evaluación (15%).

### 1.5.2. Contribuciones

Las contribuciones del presente proyecto son conocimiento nuevo y herramientas para uso en la WoT, descritas de la siguiente manera:

- El conocimiento nuevo consiste en la propuesta de un modelo de referencia, el cual establece un conjunto de lineamientos para la generación de modelos gráficos en 2D o 3D para interfaces AR en la WoT.
- En cuanto a las herramientas software tenemos el desarrollo de paquetes, denominados en esta tesis como *Plugins*, que permitan la representación de objetos inteligentes mediante modelos gráficos y sus servicios en un escenario específico, centrándose en su reutilización y personalización.



## 1.6. ESTRUCTURA DE LA MONOGRAFÍA.

El presente trabajo se encuentra organizado de la siguiente manera:

### **CAPÍTULO 1: Presentación del proyecto.**

Se presenta una introducción al tema, donde se plantea el problema que dio origen a la investigación en el que también se mencionan la justificación, los objetivos, el alcance y las contribuciones.

### **CAPÍTULO 2: Marco teórico.**

En este capítulo, se presenta la definición de los diferentes conceptos involucrados en el proyecto de investigación y se realiza una caracterización de los estudios más relevantes entorno al problema planteado, con el objetivo de proponer el modelo de referencia para representación gráfica de IO en AR para la WoT.

### **CAPÍTULO 3: Definición del Modelo de Referencia para Representación Gráfica de IO.**

Partiendo de la documentación, estudio y análisis de diferentes referencias bibliográficas, en este capítulo se presenta la definición formal del Modelo de Referencia, el cual se encuentra enmarcado por una descripción detallada de los lineamientos de los que se compone. Dichos lineamientos proporcionan los parámetros para la creación de *Plugins* que componen modelos gráficos de IO en AR para la WoT. También se muestra una prueba de concepto, donde se aplica el modelo de referencia.

### **CAPÍTULO 4: Construcción del estudio de caso.**

En este capítulo, se lleva a cabo la construcción de una interfaz de AR, para el contexto de Domótica<sup>8</sup>, la cual utiliza el modelo de referencia explicado en el capítulo anterior. La construcción de la interfaz se realiza mediante un entorno de desarrollo y una herramienta software que facilite la aplicación de la AR, teniendo en cuenta investigaciones previas acerca de las ventajas y desventajas de varias herramientas que existen en la actualidad.

### **CAPÍTULO 5: Evaluación y Análisis.**

En este capítulo se presenta la metodología de evaluación y su respectivo análisis. Además, se aplican algunos criterios de evaluación que permitirán probar la funcionalidad de los *Plugins* construidos mediante el modelo de referencia, centrándose en medir la capacidad de personalización y reutilización en diferentes contextos de aplicación.

### **CAPÍTULO 6: Conclusiones y trabajo futuro.**

Finalmente, en este capítulo se realiza las conclusiones de los resultados obtenidos en el desarrollo de la investigación y se plantean los nuevos desafíos que conducen el desarrollo de nuevas investigaciones.

---

<sup>8</sup> Conjunto de técnicas orientadas a automatizar una vivienda, que integran la tecnología en los sistemas de seguridad, gestión energética, bienestar o comunicaciones.



Adicionalmente, en los anexos se presenta información complementaria sobre los temas tratados en los capítulos, así:

**ANEXO A**

Se muestran las fichas bibliográficas seleccionadas para conformar el núcleo temático del modelo de referencia.

**ANEXO B**

Se muestran los artefactos pertenecientes al modelo de referencia, tales como el código fuente e imágenes del prototipo en Java del paso cinco y el diagrama de actividades del modelo.

**ANEXO C**

Se muestra el formato guía de lineamientos generales, como una de los artefactos principales del modelo de referencia.

**ANEXO D**

Se describen el código fuente y los recursos gráficos creados para la funcionalidad de la interfaz AR para la WoT del Termo Inteligente.

**ANEXO E**

Se describen el código fuente y los recursos gráficos creados para la funcionalidad de la interfaz AR para la WoT del escenario de Domótica.

**Anexo F.**

Se muestran los test realizados en la evaluación y análisis del modelo de referencia.

Modelo de Referencia para Representación Gráfica de Objetos  
Inteligentes en Realidad Aumentada Para la WoT.

---







## CAPÍTULO 2 MARCO TEÓRICO

### 2.1. INTRODUCCIÓN.

En este capítulo se presenta la definición de los conceptos más importantes que se involucran en todo el proyecto de investigación, además se realiza una caracterización de los estudios más relevantes entorno al problema planteado y posteriormente se detalla la construcción del modelo de referencia. El modelo de referencia establece un conjunto de lineamientos y un proceso paso a paso que permite la creación de modelos gráficos de IO, basándose en la técnica de AR para la WoT. Por lo tanto se presentan los siguientes conceptos fundamentales para su construcción.

### 2.2. CONCEPTOS FUNDAMENTALES.

Al enfocarse en el desarrollo del modelo de referencia, para representación gráfica de objetos inteligentes basándose en la técnica de realidad aumentada para la WoT, es necesario ubicar información referente a los conceptos en los que se basa la realización del actual proyecto.

#### 2.2.1. Internet de las Cosas (Internet of Things - IoT).

La Internet de las cosas se enfoca en la interconexión digital de cosas comunes, tales como: electrodomésticos, muebles o cosas tan cotidianas como un asiento. Una de las primeras cosas que se conectaron a la Internet fue una tostadora, esto con la intención de manipular el electrodoméstico sin la necesidad de una interacción física real, tan solo una interacción virtual desde otro dispositivo.

En la perspectiva de la IoT, se calcula que para el año 2020 existirán aproximadamente más de 26 mil millones de cosas conectada a la internet [18], haciendo de esta industria una de las más estables y de mayor aumento al pasar el tiempo. La IoT se propuso formalmente por Kevin Ashton en el Auto-ID Center del MIT en 1999 [19]. Desde entonces se han realizado varios estudios con respecto a la composición hardware y de protocolos de comunicación que pertenecen a la IoT. Los componentes principales de la IoT son: los sensores, los actuadores, los componentes electrónicos e interfaces de comunicación cableada e inalámbrica. La IoT es capaz de monitorear y permitir al usuario interactuar con su entorno y otros dispositivos en tiempo real. Por último, la IoT se establece entre las 6 primeras capas del modelo OSI<sup>9</sup>[20], donde se encuentra una amplia variedad de protocolos de comunicación poco integrables entre dispositivos, ya que cada uno tiene su propia marca o protocolo de interacción [21].

---

<sup>9</sup> El modelo OSI es un modelo de referencia distribuido en 7 capas, las cuales se enfocan en caracterizar y estandarizar las funcionalidades de un sistema de comunicación por medio de capas. Estas capas son: aplicación (los servicios de red a aplicaciones), presentación (representación de los datos), sesión (comunicación entre los dispositivos de la red), transporte (conexión extremo a extremo y confiabilidad de los datos), red (direccionamiento lógico con IP), enlace a datos (direccionamiento físico) y físico (señal y transmisión de datos por medio de binarios). Cada capa se comunica con la capa debajo de ella, realizando así funciones de comunicación continua entre capas y llevando los datos a donde se requieran [20] H.Zimmermann, "OSI reference model--The ISO model of architecture for open systems interconnection," *IEEE Transactions on communications*, vol. 28, pp. 425-432, 1980.

La IoT tiene variedad de aplicaciones en diversos contextos, por lo tanto, los autores Asghar, et al. [19] establecen algunos dominios de aplicación, pero a su vez tiene en cuenta las características la disponibilidad de la red, la escala, la cobertura y la repetitividad. La clasificación de aplicaciones se muestra en la Ilustración 2. Por ejemplo, se puede observar el campo de aplicación de la salud.

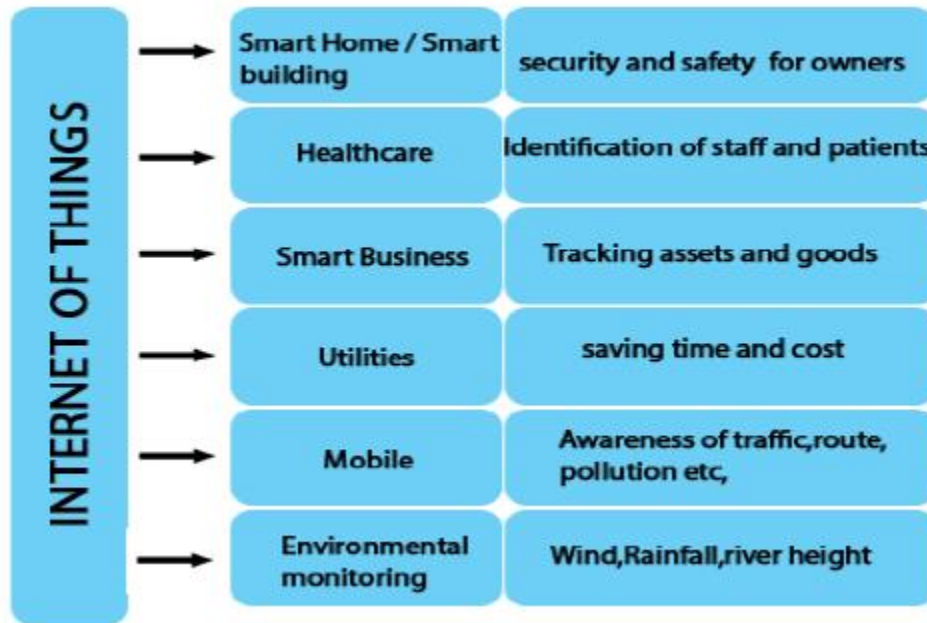


Ilustración 2 - Aplicaciones de la IoT

Fuente: Asghar, et al. [19], Principle Application and Vision in Internet of Things (IoT)

En el campo de la salud la IoT está salvando vidas, ya que a través de ella se puede monitorear a pacientes, medicinas y servicios varios, de una manera rápida y eficaz, acoplándose a las necesidades del médico y el paciente, demorando menos tiempo los procedimientos y realizando cirugías menos riesgosas y exactas. Lo anterior son algunos beneficios que otorga la IoT al campo de la medicina [22] por lo cual se observa el impacto que está teniendo esa tecnología en el ecosistema humano.

### 2.2.2. Web de las Cosas (Web of Things - WoT).

La Web de las Cosas (WoT) va de la mano con la IoT. Esto debido a que el propósito de la WoT es acortar la amplia brecha que existe en la actualidad entre el mundo real y el mundo digital. Teniendo presente que las cosas del mundo real están constantemente en conexión con el internet y se puede acceder a sus datos por medio de la web, el paso más lógico es consolidar a la Web como el entorno ideal para la comunicación y control de los objetos físicos. Esta idea está respaldada por los protocolos de comunicación que tiene la web, que se diferencia con los de la IoT, la cual tiene varios problemas de compatibilidad. La WoT ha logrado establecer una interfaz común de desarrollo que permite tener una sola forma de comunicarse con las cosas, se puede observar cómo es la comunicación de los objetos físicos con la web en la Ilustración 3.

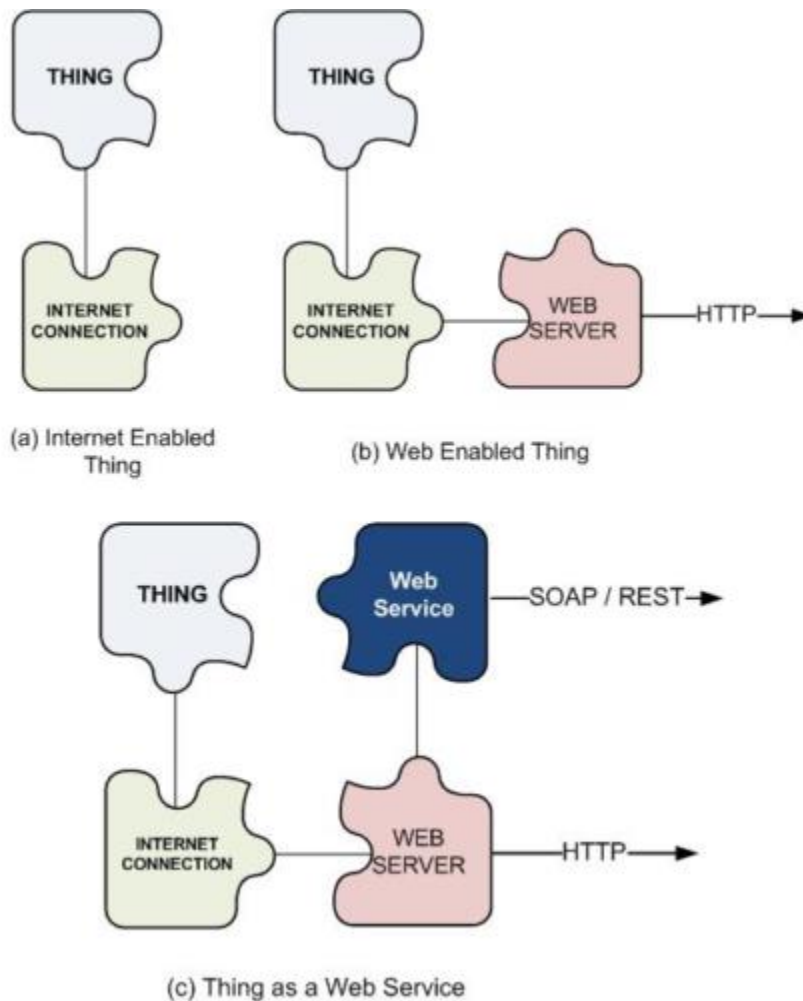


Ilustración 3 - Conectando cosas a la Web

Fuente: Mathew, et al. [4], Web of Things: Description, Discovery and Integration

Dicha comunicación se ubica en la última capa del modelo OSI, siendo esta la capa de aplicación. Lo anterior contrasta con la IoT que se compone de las primeras 6 capas del modelo OSI y donde se encuentra una gran variedad de protocolos de comunicación, interacción y componentes hardware, los cuales se producen diariamente y de manera tan diferente uno del otro que no tienen en cuenta la homogeneidad o la transparencia, es allí donde la WoT avanza y ayuda a construir una capa de aplicación para la IoT [21].

Por lo anterior se entiende que la WoT está compuesta de un solo entorno de comunicación, lo que facilita un desarrollo rápido, homogéneo y de mayor alcance a nivel de servicios para el usuario, permitiendo que los dispositivos, servicios y aplicaciones hablen entre sí a la perfección. La aplicación de la WoT en la vida real está interactuando con muchos sectores. Uno de ellos son nuestros hogares, donde encontramos múltiples cosas de las cuales se requiere estar pendiente permanentemente. Tomando como ejemplo la alacena de una cocina, donde se tiene la necesidad de estar pendiente de cosas cotidianas como la caducidad de la comida, la alimentación de la familia, los costos de llenar la alacena, etc. en este tipo de contexto la WoT en sociedad con otras técnicas para mostrar información



(como puede ser la realidad aumentada), facilita la toma y transformación de los datos para ser mostrados en dispositivos portátiles de los usuarios, generando alarmas y mejorando la calidad de vida de los individuos de la familia.

### 2.2.3. Interacción Semántica

Los objetos inteligentes en la WoT, aprovechan los lenguajes y protocolos basados en la Web para alcanzar la interoperabilidad y ofrecer a los usuarios posibilidades de interacción avanzadas. A medida que el número de objetos conectados aumenta de manera exponencial, surge una vital necesidad de proponer soluciones que faciliten la comunicación entre los objetos físicos y la interacción con el usuario. Además, en una situación típica, los objetos no son actualmente capaces de comunicarse entre sí sin un debido proceso de configuración [23]. Debido a esto, la web de las cosas incorpora cierta inteligencia en objetos conectados a internet para comunicarse, intercambiar información, tomar decisiones, invocar acciones y proporcionar servicios sorprendentes. Aquella inteligencia es denominada, interacción semántica de objetos de la WoT, entendiéndose como la capacidad que tienen los objetos inteligentes para comunicarse entre sí y proveer mejores servicios [11] en un contexto definido, mostrando el comportamiento cooperativo con el fin de resolver requerimientos de los usuarios, aplicaciones y otros objetos físicos.

### 2.2.4. Arquitectura del Proyecto.

La arquitectura que se tendrá en cuenta a lo largo de este proyecto, es la propuesta por Niño-Zambrano [11], llamada “Arquitectura de interacción semántica entre objetos inteligentes de la IoT”, esta arquitectura está orientada por medio de un modelo de interoperabilidad semántica en la WoT, que se enfoca en la interacción entre los objetos inteligentes, para proveer mejores servicios. Este modelo, involucra estándares de la IoT, infraestructuras middleware, técnicas de la web semántica y ontologías, proporcionando así inteligencia a los objetos de la WoT.

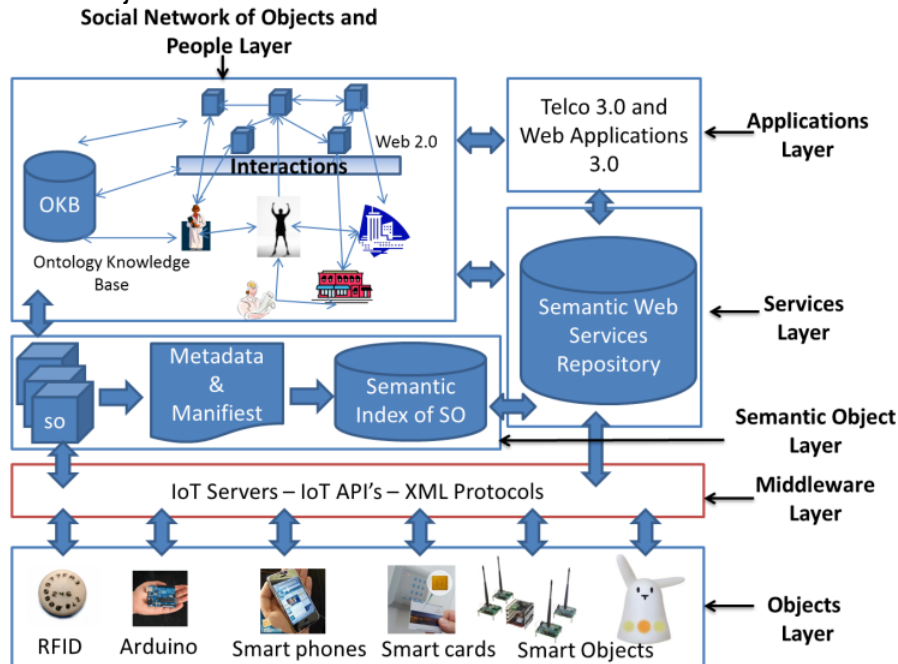


Ilustración 4 - Arquitectura de interacción semántica entre objetos inteligentes de la IoT.



### 2.2.5. Objetos Inteligentes.

Según Mathew, et al. [4], un objeto inteligente (Smart Thing) tiene capacidades de identificación, procesamiento, almacenamiento y comunicación. Esta definición se refiere a las características físicas, para su comportamiento se puede complementar con Estrada-Martinez and Garcia-Macias [24], los cuales definen un objeto inteligente que es *“capaz de detectar, procesar la información, proporcionar servicios y comunicarse con otros objetos”*. El propósito de estos objetos es brindar escenarios donde puedan interactuar de manera inteligente compartiendo sus servicios. Por otra parte Niño-Zambrano [11], extiende el concepto de objetos inteligentes al de objeto semántico, al integrar una ontología (ontología objeto semántico) para incorporar el comportamiento inteligente. Así se tendrá en cuenta el concepto de objeto semántico, al momento de establecer los objetos de la IoT que se verán involucrados en el desarrollo del presente proyecto.

### 2.2.6. Contexto y Contexto-Consciencia.

Con respecto al contexto, Dey and Abowd [25] definen el contexto como: *“cualquier información que puede ser usada para caracterizar la situación de una entidad. Siendo una entidad una persona, lugar, u objeto que es considerado relevante para la interacción entre el usuario y la aplicación*. Por lo tanto, el contexto no solo se refiere a la recopilación de datos por medio de dispositivos, sino también, a los datos obtenidos de personas, cosas y sus respectivos comportamientos en diferentes entornos [26]. Por otra parte, tenemos al autor Perera, et al. [5], quien concuerda con Dey and Abowd [25] en su definición de contexto y centra su recopilación de datos en sensores, pero complementa la definición al introducir los tipos de contexto, estos tipos de contexto son:

- **Contexto primario:** El contexto primario se basa en la información no procesada, recolectada por sensores, clasificando estos datos en cuatro categorías: la ubicación, la identidad, el tiempo y la actividad, por ejemplo: *“Identificar un usuario basado en etiquetas RFID”*.
- **Contexto secundario:** El contexto secundario toma la información generada en el contexto primario y realiza operaciones de procesamiento de datos con el fin de obtener información complementada, por ejemplo: *“identificar a un usuario de acuerdo a su comportamiento espacial a partir de las etiquetas RFID en las que se ha identificado previamente”*.

Es de gran importancia tener en cuenta el contexto a la hora de generar un modelo gráfico de los OI, debido a que este factor facilita la personalización de los servicios con los cuales las personas interactúan. Lo anterior, se ve reflejado en el concepto de sistema sensible al contexto-consciencia (context-awareness), donde Dey and Abowd [25] lo definen como: *“Un sistema es sensible al contexto si se utiliza el contexto para proporcionar información y/o servicios de interés para el usuario, donde la relevancia depende de la tarea del usuario*. Por lo tanto, el contexto-consciencia debe ser parte fundamental de la arquitectura de desarrollo, con el fin de que las aplicaciones sean flexibles y ajustables a las preferencias del usuario independientemente del contexto en el que se encuentren.

Por otra parte, Perera, et al. [5] propone el ciclo del contexto-conciencia, con el cual explica en donde se producen los datos y donde son consumidos. El autor llega a la conclusión de que existen cuatro fases para el ciclo de vida del contexto. Podemos observar estos ciclos de vida en la Ilustración 5.

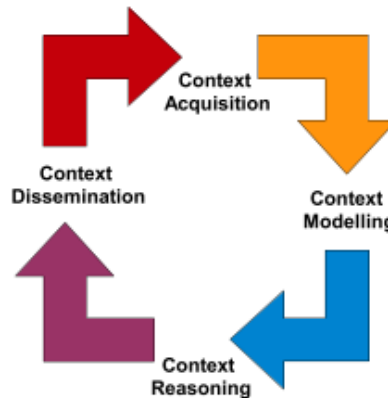


Ilustración 5 - Ciclo de vida del contexto

Fuente: Perera, et al. [5], Context Aware Computing for The Internet of Things: A Survey.

Por lo tanto, el autor plantea que el ciclo de vida está compuesto por:

- **La adquisición del contexto:** Esta adquisición, se consigue mediante técnicas para adquirir el contexto, estas técnicas se basan en la responsabilidad, la frecuencia, la fuente del contexto, el tipo de sensor y el proceso de adquisición.
- **El modelado del contexto:** Consiste en generar una representación del contexto. Esta representación del contexto, llamado modelo del contexto, se puede diseñar con modelos estáticos, el cual involucra información predefinida y con modelos dinámicos, que varían constantemente la información con la que trabajan. Para modelar el contexto se tiene en cuenta la información con la que se va trabajar y posteriormente se pasa a validar el modelo producido. Para implementar un modelo de contexto se tienen en cuenta algunas técnicas, como: el modelado llave-valor, el modelado por marcado de esquema, el modelado gráfico, el modelado basado en objetos, el modelado basado en lógica y el modelado basado en ontologías.
- **El razonamiento del contexto:** Se entiende como un “método de deducción del conocimiento”, el conocimiento se relaciona con lo que se aprendió acerca de los contextos que se tiene en cuenta. El razonamiento del contexto es medido a través de métricas, tales como: la eficiencia, la solidez o robustez, la completitud y la interoperabilidad.
- **La distribución del contexto:** Suministra métodos para entregar el contexto a los consumidores, los métodos que se utilizan comúnmente son el método de consulta y el método de distribución.

Para el presente proyecto, se tendrá en cuenta el contexto a la hora de generar y adaptar modelos gráficos para los IO de la IoT. Actualmente, en el grupo de investigación en tecnologías de la información - GTI, existe un proyecto que está modelando el contexto centrado en el usuario, a través de ontologías. El presente proyecto se basará en este modelo para acceder al contexto de usuario y adicionalmente, para el contexto del entorno de aplicación, se realizará a través del índice semántico creado en el proyecto [11].





### **2.2.7. Representaciones Gráficas en 2D.**

Un modelo gráfico en 2D está compuesto por un espacio bidimensional, el cual obtiene la representación de la información, proyectada en dos dimensiones de manera plana en el espacio físico. Actualmente las representaciones gráficas en 2D son explícitas a un usuario, sin embargo, con el paso del tiempo y los avances tecnológicos, estas representaciones se han convertido en algo más robusto, involucrando datos más amplios sobre el color, la textura, la forma, entre otros [27]. En contexto, para generar un modelo de representación gráfica de objetos WoT en dos dimensiones, existe actualmente un conjunto determinado de estándares, especificaciones técnicas y pautas las cuales establecen un dominio de escogencia en colores, textura, tamaño, forma y ubicación propuesta por la W3C [28]. En base a lo anterior, la presente investigación, buscará incluir o establecer una adecuada representación gráfica para los objetos inteligentes.

### **2.2.8. Representaciones Gráficas en 3D.**

Un modelo gráfico en 3D está compuesto por una colección de puntos conectados por líneas en un espacio tridimensional, generando modelos geométricos basados en polígonos o triángulos. Es una réplica matemática de un objeto, ya sea real o imaginario. Hoy en día se generan por medio de software dedicados a la creación de gráficos 3D, también se pueden formar de manera física por medio de impresoras 3D, a través de algoritmos o escaneados. Estas representaciones de tres o más dimensiones, generan una mejor abstracción de la información por parte del usuario. Por lo tanto, actualmente este tipo de modelos son aplicados en diversas áreas, una de ellas es la medicina, donde se usan estos modelos para observar mejor las enfermedades que a veces son difíciles de detectar, como el cáncer o para entrenar a futuros médicos por medio de simuladores basados en modelos 3D [29].

### **2.2.9. Realidad Aumentada.**

La interacción humano computador (Human Computer Interaction - HCI) se relaciona con la usabilidad de un producto y por lo tanto con su interfaz; su interfaz es fundamental para que este producto sea comunicativo. Por lo tanto, se tiene en cuenta que la interfaz es el medio en el que se entabla la primera interacción con el usuario, siendo básico que la información de los diseños visuales sean fáciles de entender [30]. La AR tiene los componentes que se buscan para una representación gráfica, con el objetivo de ser un medio por el cual se transfiere información virtual al mundo real. Existen múltiples aplicaciones de la AR en diferentes áreas, tales como: la visualización médica, el entretenimiento y la publicidad entre otras. La AR enriquece la realidad con objetos virtuales, los cuales brindan información adicional al usuario, que por medio de sus sentidos no puede adquirir [31].

Milgram y Kishino [32], definen que la realidad aumentada nace de la realidad mixta, siendo la realidad mixta una escala que va desde los elementos completamente reales a los elementos completamente virtuales, como se observa en la Ilustración 6, llamando a este concepto como "*Virtuality Continuum (VC)*". Ellos fueron los primeros en clasificar las distintas partes de la realidad mixta. En su artículo realiza una profundización acerca de cada una de ellas, siendo la realidad virtual el acercamiento más próximo a los entornos totalmente virtuales y la realidad aumentada, más cercana a los entornos reales, donde los



objetos virtuales que se antepone a esta realidad también representan información del entorno real.

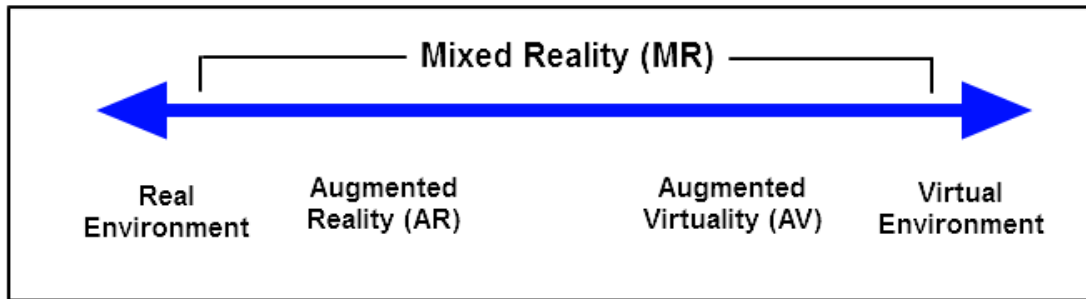


Ilustración 6 - Virtuality Continuum (VC)  
Fuente: Milgram y Kishino [32], A taxonomy of mixed reality visual displays

Por otro lado tenemos a Azuma [33], quien presenta tres componentes que debe tener la AR, estos son:

- La combinación de elementos reales y virtuales
- La interactividad en tiempo real
- La presentación es en tres dimensiones.

Así, la realidad aumentada será una parte crucial en el desarrollo del presente proyecto, debido a que la información del IO se representará de manera más cercana al usuario y la realidad en la que se encuentra.

### 2.2.10. Herramientas de Aplicación (Términos).

Guinard, et al. [34], presenta la problemática de la escalabilidad en la comunicación, debido a los miles de objetos inteligentes que en los próximos años existirán, por lo tanto en un futuro las personas estarán rodeadas de muchos IO con los que directa e indirectamente interactuarán. Por lo anterior los términos con los que se están conociendo las diferentes herramientas que extienden la funcionalidad de motores y plataformas ya establecidas, son muy variadas. Uno de estos términos lo usa Guinard, et al. [34], quien propone una arquitectura sustentada en Mashup, teniendo en cuenta que los Mashup son una forma de integrar datos de una o más fuentes, generando así la reutilización de su contenido y funcionalidad. Para el presente proyecto se tendrá en cuenta la siguiente tabla comparativa (ver Tabla 1) entre los Mashups [35], los *Plugins* [36] y *Prefabs* [37], los cuales son conocidos por extender la funcionalidad, ser reutilizables y tener propiedades de personalización en las tecnologías que los emplean esto con el fin de reconocer en esta tesis que la terminología del producto resultante cambia de acuerdo a la tecnología empleada para el desarrollo de esta.

	Características
<b>Mashups</b>	<ul style="list-style-type: none"> <li>• Va de la mano con conceptos como la integración, colaboración, reutilización y distribución de la información.</li> <li>• Capacidad dinámica de integración y reutilización de información con otras aplicaciones Web.</li> <li>• Hecho por el usuario para uso propio.</li> </ul>





	<ul style="list-style-type: none"> <li>• Orientado al consumidor.</li> <li>• La información disponible en Web y al interior de las empresas crece exponencialmente. Se necesita usar esa información para manipularla de manera rápida y sencilla.</li> <li>• Acceso rápido a la información significa mayor competitividad y productividad.</li> <li>• Se emplea solo para aplicaciones Web.</li> <li>• Utilización de servicios Web públicos.</li> <li>• Integración dinámica temporal a fuentes de información, por lo tanto la integración es concurrente.</li> <li>• La información se encuentra distribuida en diferentes fuentes de información.</li> </ul>
<b>Plugins</b>	<ul style="list-style-type: none"> <li>• Añade una funcionalidad adicional o una nueva característica al software.</li> <li>• El Plugin sea ejecutado mediante el software principal, con el que interactúa a través de una API.</li> <li>• Amplían la funcionalidad de la plataforma.</li> <li>• Facilitan la colaboración de desarrolladores externos con el software.</li> <li>• Reducir el tamaño de la aplicación.</li> <li>• Separar el código fuente de la aplicación a causa de la incompatibilidad de las licencias de software.</li> <li>• En general, cualquier aplicación puede añadir soporte para complementos.</li> <li>• Se pueden activar, desactivar o desinstalar los Plugin.</li> <li>• Puede surgir un conflicto entre un Plugin y la aplicación principal que provoque diversos fallos.</li> </ul>
<b>Prefabs</b>	<ul style="list-style-type: none"> <li>• Actualmente el termino está y sus características se usan en un entorno de desarrollo llamado Unity 3D</li> <li>• Se puede agregar componentes y propiedades que pueden ser personalizables.</li> <li>• Es reutilizable en cualquier escenario.</li> <li>• Puede ser copiado varias veces y personalizado de manera individual, sin afectar la funcionalidad de sus copias.</li> <li>• Su funcionalidad no es exportable a otros entornos de desarrollo.</li> </ul>

Tabla 1 - Tabla de Características de Términos

Como podemos observar en la Tabla 1 existen varios términos con significados similares pero que son nombrados dependiendo de la tecnología en la cual se va a desarrollar. La presente investigación busca determinar el término que mejor se acople a las características del modelo de referencia. Por lo que también se toma en cuenta la Tabla 2, para observar cual es el término más general y comúnmente usado entre los entornos de desarrollo.

Termino dependiente de la Tecnología	Herramientas y Entornos de Desarrollo
Mashups	Mapas, Videos y fotos ,APIs como RSS, REST y Web Service, aplicaciones Web Hibrida(Ionic)
<i>Plugins</i>	Navegadores Web (Mozilla Firefox, Chrome), Framework de desarrollo Web (Django, Angular), WordPress, Unity3D, Reproductores de Video y Audio
Prefabs	Unity3D



Tabla 2 - Variación de la terminología del producto dependiente de la tecnología seleccionada para el desarrollo software

Por lo anterior se encontró que el desarrollo a base de Plugin (Componente) es el término más común y sus características se acoplan al objetivo del presente trabajo de grado, como es la reutilización, personalización (con algunas limitantes) y la facilidad de colaboración de desarrolladores externos, debido a la anterior el término usado a lo largo del trabajo de grado será Plugin.

### 2.3. TRABAJOS PREVIOS.

Como trabajos relacionados se encuentran autores como G.Fortino, et al. [38] que en su investigación titulada como *"Integration of Agent-based and Cloud Computing for the Smart Objects-oriented IoT"* destacan al desarrollo de objetos inteligentes, como su gestión e integración en aplicaciones reales. Son desafíos complejos que requieren de modelos, técnicas y tecnologías adecuadas para su implementación. Concluyendo principalmente que existe una necesidad de tener actualmente buenas representaciones realizadas mediante modelos gráficos adecuados que otorguen claramente información de los datos que contienen.

En otra de las investigaciones relacionadas, se encuentra la investigación realizada por Leppanen, et al. [14] en donde hacen uso de un framework de aplicaciones AR móvil basado en web, pero todo el desarrollo de las aplicaciones móviles son hechas a la medida, haciendo uso de la realidad aumentada nombrándolas como aplicaciones MAR. En la investigación se resalta que los desarrollos rápidos de tecnologías de la IoT, han llevado a la computación los recursos disponibles al alcance de las personas. Estos recursos informáticos incluyen componentes como sensores y actuadores que mediante una interfaz de usuario enriquecida para el seguimiento y control de entornos inteligentes otorgan información digital sobre la visión del mundo real. Pero como anteriormente se dijo, son desarrollos hechos a la medida que no disponen de la aplicación al concepto de reutilización entre sus proyectos.

Otro de los trabajos relacionados a destacar es el de A.Martínez-Ospina [39] en su investigación titulada como *"Realidad aumentada en interiores: posicionamiento del usuario en dispositivos móviles y aplicaciones en rehabilitación y guiado (GuIAR)"* en la que destaca que desde el año 2009, se ha provocado un crecimiento exponencial de aplicaciones sobre la tecnología de realidad aumentada, resaltándola como una causante de nuevas brechas a la innovación. De igual manera, resalta que el uso de la realidad aumentada cada vez es mayor a medida que pasa el tiempo, y actualmente se puede apreciar su uso en diferentes campos de aplicación, como lo es: el ámbito educativo, publicitario, cultural, medicinal y entretenimiento. También comenta que de una u otra forma los dispositivos móviles han hecho que la IoT sea mucho más accesible, provocando que en un futuro tenga tal fuerza que permitan grandes avances para los seres humanos. La autora también propone un prototipo generado mediante AR para la orientación personalizada en espacios interiores, el cual utiliza los sensores disponibles en el dispositivo móvil para generar la detección de la posición en tiempo real el proyecto es bautizado como GuIAR y es orientado a dos contextos de aplicación, tales como medico y educativo.



Es de resaltar que los trabajos relacionados destacados anteriormente, son trabajos que brindan soluciones hechas a la medida, que hacen uso de tecnologías como la realidad aumentada y la IoT. Sin embargo, se debe tener en cuenta que las representaciones graficas de los objetos inteligentes mediante la realidad aumentada contienen un componente hardware dentro del cual se destacan las investigaciones de Riobamba y Guerrero [17], el cual implementa mecanismos de interacción semántica de objetos inteligentes para la WoT, que permita identificar, crear y ejecutar nuevos servicios como producto de su interacción. Este proyecto implementa un escenario de prueba basado en el MIOI (Módulo de Interacción de Objetos Inteligentes), a su vez manejan ontologías que permiten definir e implementar el mecanismo de comunicación entre de objetos inteligentes y crear nuevos servicios procedentes de esta comunicación. El escenario presentado en el anterior proyecto, será el que se usará como estudio de caso, del presente trabajo de grado al cual se le aplicará el modelo de referencia documentado a lo largo de este documento.

En general el escenario se compone de tres raspberry Pi3 equipadas con sistema operativo Linux, tres tarjetas grove, tres sensores grove (sensor de luz, temperatura, humedad) y cuatro actuadores (led, ventilador, calefactor, riego simulado por un buzzer). El escenario que se va representar gráficamente contiene tres recursos IoT considerados objetos inteligentes de la web de las cosas:

- **Regulador de Luz:** Contiene un sensor de iluminación y una lámpara, otorgando el servicio de activar la lámpara cuando el sensor no capte nada de luz.
- **Regulador de Temperatura:** Contiene un sensor de temperatura, un actuador calefactor (emulado por un secador de aire caliente) y un actuador ventilador, este regulador otorga el servicio de activar el calefactor en caso de que la temperatura se encuentre por debajo de 15 grados centígrados y en caso contrario, cuando el sensor capte temperatura mayor a 30 grados centígrados, se active el actuador ventilador.
- **Regulador de Humedad:** Contiene un sensor de humedad y un dispositivo de riego (emulado por un buzzer).

Los anteriores objetos inteligentes contienen distintos sensores y actuadores, que a su vez tienen servicios los cuales se van a representar gráficamente aplicando la técnica de realidad aumentada.

Estrada-Martinez and Garcia-Macias [24] En su trabajo de investigación titulado “*Semantic Interactions In The Internet Of Things*” muestra distintos escenarios inteligentes, definiendo tres paradigmas de la IoT entre los cuales son: orientados a la semántica, al internet y a las cosas, resaltando la semántica como el proceso de investigación sobre la representación, almacenamiento, búsqueda y organización de la información generada por los IO en espacios físico denominados escenarios inteligentes.

Por otro lado, el proyecto, desarrollado por Guzmán-Álvarez [16], genera una interfaz en realidad aumentada, para el escenario de interacción semántica permitiendo identificar un contexto definido y adicionalmente gestionar y crear servicios de interacción en la Internet de las cosas. Sin embargo, también recalca las dificultades como lo es la falta de una

## Modelo de Referencia para Representación Gráfica de Objetos Inteligentes en Realidad Aumentada Para la WoT.

---



representación gráfica adecuada de los IO en el escenario AR creado, ya que los datos de los IO, fueron representados con modelos primitivos (esferas, cubos, etc.). Pero muestra un enfoque metodológico guiado por prototipos ideal para el desarrollo de proyectos orientado a la AR el cual se tendrá en cuenta en el presente trabajo de grado.



## CAPÍTULO 3 MODELO DE REFERENCIA PARA REPRESENTACIÓN GRÁFICA DE IO

A partir del marco teórico propuesto en el capítulo anterior, se procede a la creación del modelo de referencia el cual está integrado por dos componentes:

- Definición del conjunto general de lineamientos para la creación de modelos gráficos de objetos inteligentes en realidad aumentada.
- Definición formal del modelo de referencia.

Para la definición del conjunto de lineamientos se tiene en cuenta las características de los objetos inteligentes, tales como: dispositivos, sensores, actuadores y servicios. También se procura definir mecanismos de personalización<sup>10</sup> y reutilización para las representaciones en AR, siempre teniendo en cuenta el contexto de aplicación. Para la definición formal del modelo se proponen pasos a seguir para generar *Plugins* que permitan la construcción de las características anteriormente descritas.

### 3.1. DEFINICIÓN DEL CONJUNTO GENERAL DE LINEAMIENTOS PARA LA CREACIÓN DE MODELOS GRÁFICOS DE OBJETOS INTELIGENTES EN REALIDAD AUMENTADA

Los estándares, recomendaciones técnicas y pautas que otorga la W3C, buscan mostrar gráficamente información explícita al usuario, la cual contenga las características de ser una información clara<sup>11</sup>, oportuna<sup>12</sup> y concisa<sup>13</sup>. Si bien el crear una representación en 3D o 2D, implica mucha creatividad, se puede orientar algunas características de dicha representación.

Debido a lo anterior, se toman las siguientes definiciones del marco teórico: objeto inteligente, representación gráfica en 2D, representación gráfica en 3D y los conceptos de la W3C, como base de conocimiento para plantear un formato guía el cual establece los componentes que pueden contener las diferentes representaciones gráficas de los IO por medio de la técnica de realidad aumentada y de acuerdo a las características de sensores, actuadores y servicios que los componen.

A continuación se detalla en la Ilustración 7 el formato guía para la definición del conjunto de lineamientos, los cuales serán llamados lineamientos generales y se tendrán en cuenta a lo largo de la definición del modelo y en la creación de las representaciones gráficas de los objetos inteligentes en realidad aumentada. El formato busca orientar al desarrollador en la creación o reutilización de los *Plugins* que representen los IO según sea el contexto de su escenario.

<sup>10</sup> Teniendo en cuenta que los indicadores de personalización tratados en la presente tesis son el tamaño, la textura o el color, que pueden ser variables en los modelos gráficos.

<sup>11</sup> Clara: Se refiere al tipo de información presentada al usuario como aquella información que se otorga de una manera transparente, por medio de que cualquier usuario logre entenderla.

<sup>12</sup> Oportuna: Se refiere al tipo de información que se otorgue al usuario de manera eficaz y cuando el usuario la requiera.

<sup>13</sup> Concisa: Se refiere a que se debe otorgar la cantidad de información netamente necesaria.



FORMATO GUÍA DE LINEAMIENTOS GENERALES PARA LA REPRESENTACIÓN DE IO EN AR									
NOMBRE IO	NOMBRE SENSOR O ACTUADOR	SERVICIO - Descripción del servicio	CONTEXTO		PERSONALIZACIÓN	REPRESENTACIÓN EXISTE O NO			
			Tipo de Dato	Entidad de Interés					
Se considera objeto inteligente, todo aquel que sea capaz de detectar, procesar la información, proporcionar servicios y comunicarse con otros objetos.  Aquí se ingresa el <b>Nombre</b> del Objeto Inteligente y el <b>ID</b> del IO.	Nombre del Sensor1	Nombre del Servicio1	<b>NOTA:</b> Este espacio es para seleccionar el tipo de dato usado por el servicio: Raw ó Procesado. Teniendo en cuenta que: -Raw: Es un dato(s) plano consumido directamente del sensor. -Procesado: Es un dato(s), con un contexto secundario, ya que al dato se le aplica una manipulación al procesarlo.	<b>NOTA:</b> Una entidad es una persona, lugar, u objeto que es considerado relevante para la interacción entre el usuario y la aplicación en sí misma. En este caso el enfoque de los datos del sensor u actuador.	<b>NOTA:</b> En este espacio se ingresa una lista enumerando los indicadores que se pueden personalizar. Este espacio es para describir la entidad de interés del sensor u actuador no en el repositorio.	SI			
	Nombre del Sensor2	Nombre del ServicioN				NO			
	...					...			
	Nombre del SensorN								
	<b>NOTA:</b> Este espacio es para escribir el nombre técnico del sensor.								
	<b>ACTUADORES</b>								
	Nombre del Actuador1	Nombre del Servicio1				Tipo ...	....	....	....
	Nombre del Actuador2	Nombre del ServicioN				Tipo ...			
	...								
	Nombre del ActuadorN								
<b>NOTA:</b> Este espacio es para escribir el nombre técnico del actuador.									

Ilustración 7 - Formato Guía de Lineamientos Generales para crear representaciones gráficas de los IO de acuerdo a las características de sensores, actuadores y servicios que los componen.



De acuerdo a la descripción del anterior formato, se genera un ejemplo en la Ilustración 8 para un solo sensor y actuador de la prueba de concepto aplicada a un *Termo Inteligente*, como una propuesta para describir los componentes necesarios en su representación gráfica teniendo en cuenta la información brindada por los IO del Termo y las características de personalización que se desean tener.

FORMATO GUÍA DE LINEAMIENTOS GENERALES PARA LA REPRESENTACIÓN DE IO EN AR						
NOMBRE IO	NOMBRE SENSOR O ACTUADOR	SERVICIO - Nombre métodos	CONTEXTO		PERSONALIZACIÓN	REPRESENTACIÓN EXISTE O NO
			Tipo de Dato	Entidad de Interés		
<b>SENSORES</b>						
Nombre: Termo Inteligente. ID: 7539521852	Nombre Sensor1: Temperatura DS 1820	Servicio1: Obtener dato del sensor de temperatura.	Raw	Temperatura del liquido del termo	Se puede obtener el dato de dos formas: -Por medio de MQTT. -Por medio de API REST.	NO
		Servicio2: Mostrar la temperatura por medio de una representación en 3D, con la técnica de AR.	Raw	Temperatura del liquido del termo	Se realiza una representación del liquido, por medio de un modelo 3D, el cual es modificable en los siguientes aspectos: - Textura: Se selecciona la textura o rango de texturas que se desean visualizar, de acuerdo a la temperatura del sensor en un modelo 3D. - Tamaño: Se puede modificar la escala sin deformar el modelo.	NO
<b>ACTUADORES</b>						
	Nombre Actuador1: LED RGB	Servicio1: Obtener estado del actuador.	Raw	Estado activo o desactivo del LED	Se puede obtener el dato de dos formas: -Por medio de MQTT. -Por medio de API REST.	NO
		Servicio2: Mostrar el estado del LED por medio de una representación en 3D, con la técnica de AR.	Raw	Estado activo o desactivo del LED	Para establecer actividad o inactividad del actuador se establece una representación 3D con forma de señal, a la que se podrá modificar el color de la señal.	NO

Ilustración 8 - Ejemplo instanciado del Formato Guía de Lineamientos Generales para los IO de un Termo Inteligentes





Una vez llenado el formato guía para establecer los lineamientos generales de las representaciones gráficas de los IO de acuerdo a las características de los sensores, actuadores y servicios que los componen, junto a los mecanismos de personalización y reutilización, se procede a realizar la propuesta para crear los *Plugins* en AR que representen gráficamente los IO mediante el uso de la definición formal del modelo de referencia consignado en este trabajo de grado.

### **3.2. DEFINICIÓN FORMAL DEL MODELO DE REFERENCIA.**

La definición formal del modelo de referencia se realiza mediante una metodología [40], la cual implementa tres fases, donde la fase inicial determina el ámbito del modelo de referencia, la fase dos o intermedia define formalmente el modelo de referencia y la fase tres o fase final emplea el diseño de la prueba preliminar del modelo construido. Se tiene en cuenta que esta definición formal busca establecer lineamientos específicos, adicionales al conjunto de lineamientos generales establecidos en la Ilustración 7

#### **3.2.1. Fase 1: Ámbito del modelo de referencia**

En la actualidad, las personas capturan mejor la información mediante artefactos gráficos, esto hace que las representaciones gráficas de cualquier tipo de información aporten un mejor entendimiento y percepción de las cosas que los rodean. En cuanto al acople de la tecnología, la realidad aumentada y el internet de las cosas orientado a servicios de la web de las cosas, apunta a tener un entorno capaz de desarrollar modelos gráficos con una descripción detallada de datos. Estas tecnologías, actualmente se están integrando a las personas por medio de dispositivos de fácil acceso y de uso cotidiano como son los teléfonos inteligentes, al soportar la representación de modelos 3D o 2D, que sean explícitos al ser humano, aportando a las personas un soporte más amplio en la toma de decisiones de su vida cotidiana.

En contexto, si una persona desea visualizar el clima actual a las afueras de su hogar y consecutivamente toma una decisión de acuerdo a la información brindada, deberá dirigirse a su dispositivo electrónico y consultar datos planos que muestran información del estado del clima; sin embargo, estos datos otorgados por objetos inteligentes, son datos textuales que no refieren nada explícito gráficamente.

Para una mejor descripción de los datos, existen soluciones técnicas que buscan representar gráficamente los datos de IO mediante la técnica de AR. Actualmente, existen estudios que abarcan las representaciones gráficas de datos de los IO mediante la técnica de AR. Aunque dichos proyectos son hechos a la medida y no poseen la característica de ser modelos gráficos reutilizables ni personalizables en otros contextos de aplicación. Debido a esto, se percibe la ausencia de modelos con características para representar adecuadamente los objetos inteligentes en la WoT.

Por lo tanto, se genera la importancia de contar con un repositorio común de modelos gráficos personalizables, que puedan ser implementados en soluciones con AR en la WoT. Con el fin de tener una representación unificada con la información, minimizando costos en el desarrollo y logrando su adaptación en otros escenarios. Para otorgar una solución, se plantea la creación de un Modelo de Referencia el cual se define en esta tesis como





**ARMoWoT** (Augmented Reality Reference Model for the Web of Things). ARMoWoT es una guía para generar representaciones adecuadas en 3D o 2D de los datos y metadatos que contienen los objetos inteligentes, expresados mediante la técnica de realidad aumentada, aportando a los usuarios desarrolladores un modelo basado en la reutilización de código fuente, recursos multimedia, modelos 3D, entre otros componentes, para generar aplicaciones que representen a los sensores, actuadores y servicios simulados o físicos (componentes de un IO), independientes a su contexto.

Para la creación del modelo de referencia (ARMoWoT – “Modelo de referencia en realidad aumentada para la web de las cosas”), se llevó a cabo el estudio de documentos científicos, referencias web basadas en estudios científicos, referencias bibliográficas, tesis, y demás documentos de investigación enfocadas a la WoT. Lo anterior se puede referenciar en el ANEXO A del presente documento, el cual tiene en cuenta las definiciones de: modelos gráficos en 3D, la representación gráfica de objetos inteligentes, la realidad aumentada, la definición de objeto inteligente, junto con desarrollos orientados a la representación gráfica de la IoT, sus servicios y demás definiciones que se consideran relevantes para la construcción del modelo.

En el Anexo A y el marco teórico de la presente tesis, se describen detalladamente un conjunto de definiciones nombradas anteriormente, los cuales permiten establecer la base de conocimientos que facilitaran el proceso de creación del Modelo de Referencia, el cual servirá como base para el desarrollo de otros modelos que representen gráficamente los diferentes objetos inteligentes existentes mediante la técnica de realidad aumentada enfocados a las WoT. Todo basado en estudios de diferentes documentos científicos relacionados al presente trabajo de grado, con el objetivo de analizar las características que otorga la realidad aumentada en los diferentes tipos de soluciones, observando que la mayoría de las investigaciones son desarrollos hechos a la medida en base a la web de las cosas y que la única investigación realizada por G. Fortinino, et al. [38], contemplaba un desarrollo reutilizable generando un marco basado en servicios web depositados en la nube, permitiendo reutilizar dichos servicios en desarrollos para aplicaciones móviles con objetos inteligentes, sin embargo no todos los documentos científicos, contemplaban desarrollos por medio del uso de la realidad aumentada a excepción de los investigadores A Martínez-Ospina [39], O. E. Guzman-Álvarez [16], Leppanen, et al. [14], los cuales si hacen uso de la realidad aumentada en sus investigaciones, demostrando que es una de las tecnologías cada vez mas aceptada al momento de mostrar la información que otorgan los diferentes objetos inteligentes de la web de las cosas.

Concluyendo bajo el análisis de la literatura, que existe la necesidad de mostrar la información de los objetos inteligentes de la WoT a través del uso de la realidad aumentada creando soluciones que sean reutilizables y con cierto grado de personalización para que estas representaciones de la información se adecuen a diferentes contextos de aplicación

### **3.2.2. Fase 2: Definición formal de ARMoWoT**

A partir del estudio de definiciones y el análisis de la literatura de documentos científicos realizada en la anterior fase, se lleva a cabo la definición formal del modelo de referencia, el cual detallará un conjunto lineamientos, mecanismos de reutilización y personalización en cuanto a tamaño, color o textura y una serie de buenas prácticas, recomendadas a la hora de generar modelos gráficos que representen la información que otorgan los diferentes objetos inteligentes de la web de las cosas.



El modelo de referencia ARMoWoT contiene una guía para la creación de *Plugins* centrándose en la reutilización y personalización de las representaciones gráficas para los objetos inteligentes. Para generar la interfaz AR, el desarrollador deberá tener en cuenta las pautas, recomendaciones y buenas prácticas que se encuentran inmersas en el modelo de referencia.

### **3.2.2.1. Definición de los pasos para el modelo de referencia ARMoWoT**

Para conocer a profundidad cada paso y los lineamientos contenidos en los pasos (mediante actividades o sugerencias), se presenta a continuación un diagrama de flujo (ver en la Ilustración 9), que permite comprender la dinámica del modelo.

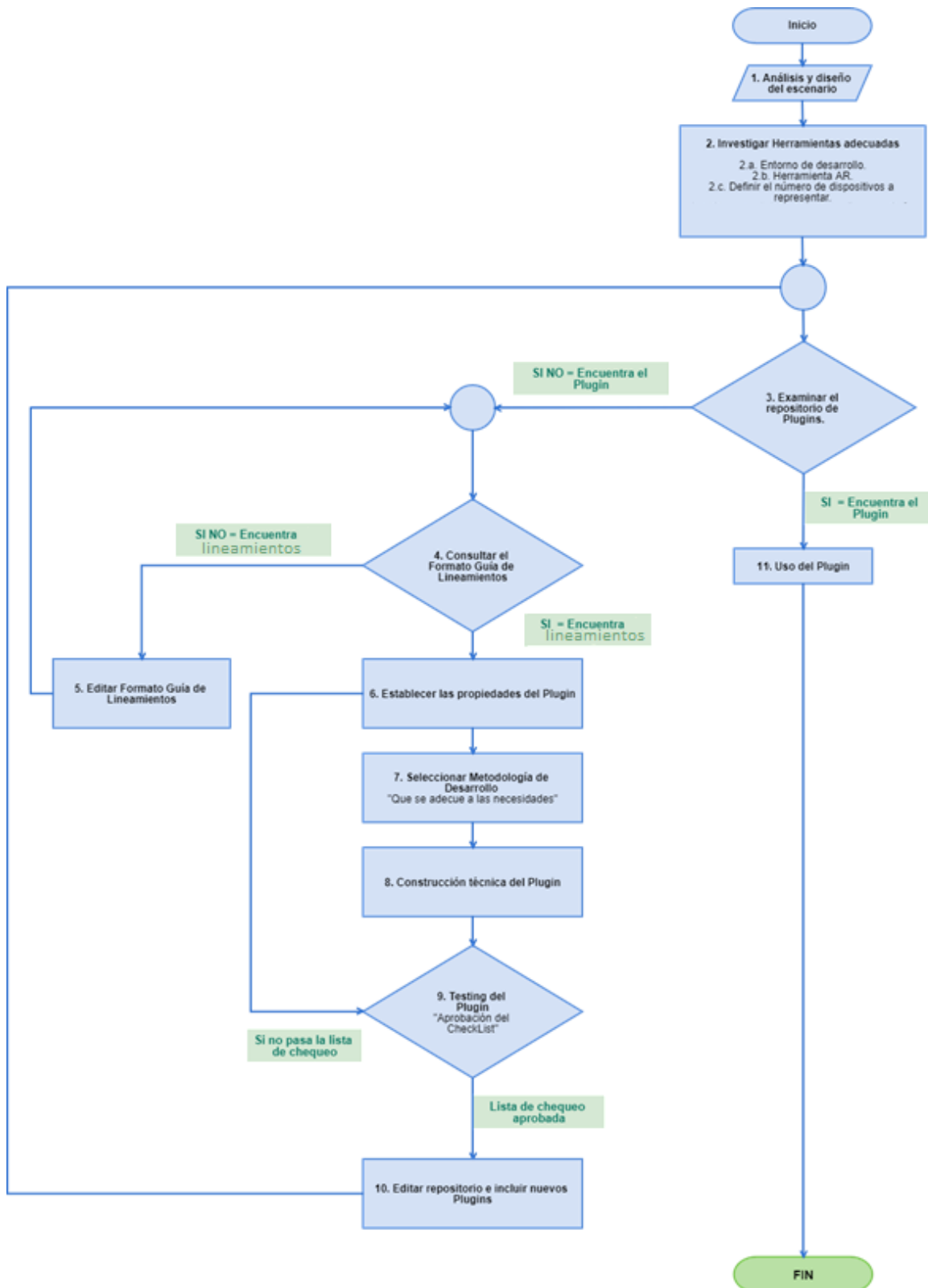


Ilustración 9 - Diagrama de Flujo para el Modelo de Referencia ARMoWoT



A continuación se detallan los pasos que enmarcan el modelo de referencia, los cuales permitirán la creación de *Plugins* que representan gráficamente los IO mediante realidad aumentada. Dichas representaciones pertenecen al grupo de datos que obtienen los diferentes objetos inteligentes independientemente del contexto de aplicación en el cual se encuentren.

Las representaciones graficas generadas con base en los lineamientos, internamente tendrán la creación de *Plugins*, los cuales deberán cumplir con las características de personalización y reutilización en cualquier contexto.

### **PASO 1: Análisis y diseño del escenario**

Como punto inicial, el desarrollador debe proponer en un párrafo una descripción general del escenario, sus requerimientos principales y el producto que desea obtener con el proyecto AR, teniendo en cuenta siempre el contexto de aplicación. Para ayudar a estructurar el párrafo se proponen los siguientes lineamientos por medio de preguntas y sub preguntas, las cuales serán respondidas en el escrito.

#### **Lineamientos específicos propuestos:**

- **¿Cuál es el escenario de aplicación?**
  - ¿Que objetos inteligentes voy a representar?
  - ¿Cuales son los componentes hardware que voy a utilizar?
- **¿Cuáles son los requerimientos más relevantes del proyecto?**
  - ¿Cuantos sensores y actuadores voy a utilizar para todos los objetos inteligentes del escenario?
  - ¿Qué servicios otorgarán los objetos inteligentes quiero representar?
- **¿Qué producto final dese obtener?**
  - ¿Como deseo obtener la representación de mi escenario?

Las anteriores preguntas, serán los lineamientos específicos del paso, sin embargo con el objetivo de especificarlos aun mas, se componen de sub preguntas, las cuales establecerán un claro análisis y buen diseño del escenario a representar.

### **PASO 2: Investigar herramienta adecuada**

El presente paso está enmarcado por 4 tareas (2.A, 2.B, 2.C, 2.D), descritas a continuación, las cuales son consideradas esenciales al momento de crear modelos para representación gráfica de objetos inteligentes en realidad aumentada:



### **PASO 2.A.: Seleccionar entorno de desarrollo**

Realizar un estudio técnico del entorno de desarrollo que se va a seleccionar para la creación de los modelos gráficos, con el objetivo de observar los beneficios, características principales, acople con realidad aumentada, IoT (Accesorios hardware – librerías para su uso en el software) y sincronización con la mayoría de protocolos IP (MQTT, HTTP & REST, CoAP, entre otros).

Para el presente paso, se realizó un estudio completo de las características más esenciales de los entornos de desarrollo más usados (el estudio está contenido en el ANEXO B), permitiendo generar una tabla comparativa, la cual sirve como apoyo para la selección del entorno que más otorgue beneficios dependiendo del escenario que se plantea inicialmente.

En el mercado de software y en la búsqueda de entornos de desarrollo se determina que una curva de aprendizaje es alta cuando el aprendizaje del uso de la nueva herramienta es difícil o contiene un proceso largo, por lo tanto para seleccionar el entorno de desarrollo ideal se debe tener en cuenta los siguientes lineamientos.

#### **Lineamientos específicos propuestos:**

- Curva de aprendizaje baja o en su defecto media.
- Que se pueda exportar a varias plataformas, principalmente a móviles (como Android o iOS).
- Que soporte protocolos de comunicación con dispositivos de la IoT (tales como MQTT, HTTP & REST, CoAP, entre otros).
- Que soporte implementación de la técnica de Realidad Aumentada o en su defecto que sea de fácil acople con herramientas AR.

### **PASO 2.B.: Seleccionar herramienta de realidad aumentada**

En este paso se realiza un estudio técnico de varias herramientas para usar la técnica de realidad aumentada, con el objetivo de seleccionar la mejor herramienta que se adapte al entorno de desarrollo seleccionado en el paso anterior. Por lo tanto se genera una tabla comparativa, a partir del estudio de diferentes herramientas de realidad aumentada, sirviendo como apoyo para establecer los siguientes lineamientos del paso.

#### **Lineamientos específicos propuestos:**

- Nivel de acople con el entorno de desarrollo seleccionado en el anterior paso.
- Curva de aprendizaje baja o media en su defecto.
- Que la herramienta tenga un modelo de negocio gratis, ya si se desea comercializar los *Plugins*, lo mejor es usar la licencia para uso comercial.
- Que la herramienta proporcione varios tipos de identificación AR (como códigos QR, imagen sola, detección en objetos físicos reales, etc.).

Por último se recomienda leer los cuadros del ANEXO B, donde está la investigación técnica de entornos de desarrollo y las herramientas para AR.

### PASO 2.C.: Definir número de representaciones a implementar

Una vez seleccionado el entorno de desarrollo y la herramienta para realidad aumentada, se debe definir el número de representaciones gráficas, lo primero es saber el número de sensores, actuadores y servicios que contienen los IO de su escenario.

Por lo tanto cada sensor, actuador y servicio es representado gráficamente por un *Plugin*, lo que significa que si un objeto o producto inteligente<sup>14</sup>, contiene más de un sensor, actuador y servicio, el número de dispositivos que componen al objeto inteligente corresponde con el número de *Plugins* a crear. Debido a lo anterior se establecen dos lineamientos.

#### Lineamientos específicos propuestos:

- Como se observa en el ejemplo de la Ilustración 10, se calcula el número de sensores y actuadores, para obtener el número de *Plugins* a necesitar, para la representación del objeto inteligente, en este caso 4 *Plugins*.
- Si el desarrollador desea generar más de una representación por cada componente del objeto inteligente, es una buena práctica que se tenga en cuenta las representaciones de más. Como por ejemplo representar la información de un sensor por medio de una interfaz en 2D y un modelo en 3D, para lo cual se requeriría de dos *Plugins*.

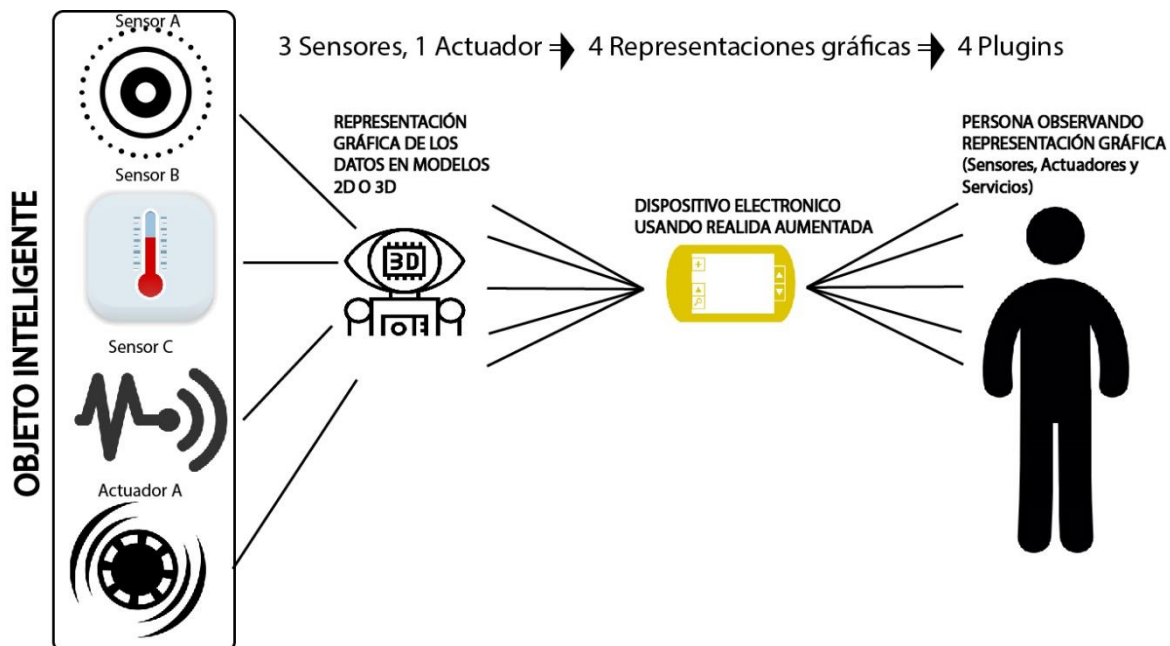


Ilustración 10 – Esquema de sensores, actuadores y servicios que componen un objeto inteligente.

### PASO 3: Examinar el repositorio de *Plugins* creados

Se sugiere el uso del repositorio creado en el presente trabajo de grado, con el objetivo de simplificar trabajo y ahorrar tiempo. Esto permite revisar en concreto, si en algún

<sup>14</sup> Para esta tesis se considera producto inteligente, cuando la entidad de interés es un producto como un termo, una mesa, una materia, lo que difiere de un contexto de domótica, contaminación ambiental, salud, entre otros.



entorno y herramienta específica de desarrollo, ya se creó el Plugin que representa el sensor, actuador o servicio de los objetos inteligente que se requieren para el escenario a representar.

A continuación se describe el link del repositorio, en el cual están almacenados todos los *Plugins* diseñados e implementados.

<https://bitbucket.org/account/user/ariotteam/projects/ARMOWOT>

En este punto se realiza un condicional de pasos, el cual describe el paso a seguir, denotando dos caminos:

1. Si el desarrollador contiene en su escenario o proyecto planteado uno o más de un dispositivo que requiere representar, entonces consulta el repositorio de *Plugins* el número de veces correspondientes al número de dispositivos (sensores, actuadores, servicios). En caso de que encuentre creado la representación de todos los dispositivos que necesita, el paso a seguir es el uso de los *Plugins* (**Paso 11**). Al realizar el uso de los *Plugins*, se prueba la capacidad de reutilización que contienen, ya que se debe adaptar bien al contexto al cual se necesita, además se verifica su capacidad de personalización, permitiendo modificarlos en tamaño, color o textura dependiendo de las preferencias del usuario, por último se sugiere un lineamiento.

**Lineamientos específicos propuestos:**

- Tener en cuenta la Ilustración 7 que es el formato guía de lineamientos generales para crear representaciones gráficas de los IO de acuerdo a las características de sensores, actuadores y servicios que los componen.
2. Ahora bien, si el desarrollador consulta el repositorio de *Plugins* y por el contrario, encuentra solo uno, algunos o ninguno de los *Plugins* que representa los dispositivos que necesita para su escenario. Entonces procede a realizar el (**Paso 4**).

**PASO 4: Revisar formato guía de lineamientos generales para representaciones graficas de los IO**

El objetivo de la revisión en la Ilustración 77, es exclusivamente para que el desarrollador tenga en cuenta como generar adecuadamente las representaciones gráficas de los sensores, actuadores y servicios de los IO, orientándolo en la creación de *Plugins* reutilizables y personalizables que representen los componentes de los IO según sea necesario en su escenario. Sin embargo, es de aclarar que aunque no haga parte del enfoque del trabajo de grado, las representaciones gráficas que genere cada desarrollador basados en el modelo, serán siempre distintas por el hecho específico de que deberán cumplir las expectativas de un usuario final y de como este usuario final desea observar información que sea explicita para el.

En este punto, se realiza un condicional de pasos, denotando dos caminos:





1. Una vez el desarrollador procede a realizar la revisión del formato guía, deberá verificar si existen lineamientos ya planteados para los componentes (sensores, actuadores y servicios) que necesite en su escenario. En caso de encontrar todos los lineamientos que necesita para crear la representación de sus componentes, deberá proceder a realizar el (**Paso 6**).
2. Si por el contrario, el desarrollador realiza una búsqueda en la Ilustración 7 y no encuentra lineamientos previamente declarados para los componentes de objetos inteligentes que necesita en su escenario, entonces deberá proceder a realizar el siguiente paso (**Paso 5**) descrito a continuación:

### **PASO 5: Retroalimentar el formato guía de lineamientos generales para representaciones graficas de IO**

En este punto se realiza una retroalimentación al modelo, el cual se vuelve más robusto con la contribución que realizan los desarrolladores y sus escenarios propuestos, esto contribuye a ampliar el contenido del modelo haciendo partícipes a los usuarios. En contexto el desarrollador debe cumplir con el siguiente lineamiento.

#### **Lineamientos específicos propuestos:**

- Diligenciar el formato de la Ilustración 7 que indica lineamientos a nivel general en este modelo, para cada uno de los componentes de los IO como sensores, actuadores y servicios que hacen parte del escenario a representar, deberá diligenciar e incluir el total de componentes que se desean representar.

Para este proceso se desarrolló una aplicación de escritorio como primer prototipo (ver en el ANEXO B), para que el formato sea fácil de llenar y se encapsule en un archivo con extensión XML, esto para iniciar un trabajo futuro donde se pueda realizar búsquedas rápidas del formato.

Una vez diligenciado el formato general de lineamientos, se realiza una verificación para determinar si el escenario a representar contiene más objetos inteligentes que aun no tengan lineamientos declarados. En caso de que el escenario si contenga más objetos a representar y que su vez no estén contenidos en el formato guía, se diligencia el formato hasta que los lineamientos de todos los objetos del escenario estén totalmente declarados.

En caso de que el formato guía ya contenga todos los lineamientos que requiere para todos los componentes, deberá proceder al (**Paso 6**) el cual se describe a continuación.

### **PASO 6: Establecer propiedades del Plugin**

En este punto el desarrollador ya encontró en el formato guía los parámetros para crear representaciones gráficas de los IO de acuerdo a las características de sensores, actuadores y servicios que los componen. Ahora se procede a establecer las propiedades del Plugin que se va a crear, el cual va a representar un componente del objeto inteligente, es decir el número de *Plugins* es respectivamente al número de componentes de los IO.





En la definición de las propiedades del Plugin, se tiene en cuenta tres aspectos considerados como lineamientos en este paso.

**Lineamientos específicos propuestos:**

- Nombre del Plugin.
- Breve descripción del problema a solucionar.
- Breve descripción de la solución.

A continuación, se presenta una guía (ver Tabla 3) para establecer las propiedades de los *Plugins* que se van a crear:

Metadatos de un Plugin		Propiedades Del Plugin		
		Nombre de los <i>Plugins</i>	Descripción del problema a solucionar	Descripción general de la posible solución
<b>PLUGINS</b>	<b>Nombre de los metadatos del Plugin</b> Puede ser el nombre físico del sensor o actuador.	Plugin ____	Se requiere crear un modelo que represente el componente sensor <u>X</u> el cual mide <u>Y</u> de <u>Z</u> objeto.	Se selecciona el entorno de desarrollo <u>A</u> y la herramienta <u>B</u> para la creación de AR, en base al estudio de características y beneficios que proporcionan a la hora de crear productos con las características de un Plugin. Por lo tanto se representa gráficamente el objeto <u>Z</u> . Estos objetos inteligentes arrojan datos planos mediante un protocolo de comunicación (HTTP, JSON, MQTT, REST, etc..), Los datos son entregados al entorno de desarrollo ____, una vez el entorno obtenga los datos de los dispositivos <u>X</u> , mediante librerías y la herramienta AR ____ seleccionada, se realiza la creación del Prefabricado (Plugin). Como resultado final se obtiene mediante AR la representación gráfica de los datos que arrojan los diferentes dispositivos que contienen el objeto inteligente.
	<b>Nombre de los metadatos del Plugin</b>	Plugin ____	Se requiere crear un modelo que represente el componente actuador <u>X</u> el cual representa <u>Y</u> de <u>Z</u> objeto.	

Tabla 3 - Metadatos de los *Plugins*

Una vez se establezcan las propiedades del Plugin se procede al **(Paso 7)**.

**PASO 7: Seleccionar metodología de desarrollo**

En este paso se lleva a cabo la selección de una metodología de desarrollo que se adecue a las necesidades de la creación de *Plugins* que complementaran el proyecto o escenario a desarrollar.



Apoyándose en el estudio que realizó Guzmán-Álvarez [16] se tiene una tabla (contenida en el ANEXO A) donde se muestra las metodologías de desarrollo y las tecnologías implementadas para crear aplicaciones que usan realidad aumentada. Por lo tanto el autor concluye que existe variedad de metodologías utilizadas en proyectos AR, entre ellas predomina el uso de prototipos y escenas debido a que el entorno de desarrollo junto a la herramienta AR seleccionada no es especializado, si no que requiere de una adaptación para interactuar con las funcionalidades que brindan.

Este estudio realizado soporta la idea de que una correcta selección del entorno y herramienta AR para el desarrollo de los escenarios generan grandes beneficios al desarrollador, sin embargo da idea en cuanto a las metodologías de desarrollo utilizadas, indicando que aproximadamente un 40% de desarrolladores, prefieren la selección de metodologías ágiles como Scrum y XP las cuales son adaptables y flexibles, sin embargo un 80% de desarrolladores, realizan sus proyectos mediante metodología basada en prototipos, posicionándola como metodología más viables al momento de generar cualquier tipo de desarrollo o implementación de proyectos, que contenga interfaces de realidad aumentada en la IoT.

Sin embargo, como se pretende que la creación de modelos gráficos basados en *Plugins* sea rápida, ágil y creados en cualquier entorno y/o herramienta AR, las metodologías ágiles como Scrum, XP o en su defecto combinadas, como se puede ver en el cuadro comparativo de Guzmán-Álvarez [16] llamado "*Elementos de las metodologías Scrum y XP*" (contenido en el ANEXO A), pueden proporcionar beneficios de adaptabilidad y flexibilidad al momento de desarrollar los *Plugins*. Por último se recomienda los siguientes lineamientos.

**Lineamientos específicos propuestos:**

- Seleccionar la metodología de prototipos para desarrollar un proyecto a gran escala para interfaces AR para la WoT.
- Seleccionar metodologías ágiles como Scrum y XP para la creación de *Plugins*.
- Claro está no se descarta en absoluto la selección propia o libre albedrío del desarrollador si desea escoger cualquier otro tipo de metodología de desarrollo.

**PASO 8: Creación técnica del Plugin**

En este punto se realiza la creación técnica del Plugin teniendo en cuenta las características de personalización y reutilización independiente del contexto en el cual se encuentren.

- Ejemplo: Personalización → Cambio de tamaño o color o textura.
- Ejemplo: Reutilización → Uso del Plugin en cualquier contexto.

Es de resaltar que la estética de las animaciones y su funcionalidad, son algo subjetivos a la hora de medir o determinar que son modelos representativos óptimos, ya que dependen exclusivamente de la experiencia técnica de cada desarrollador. Es decir, el modelo proporciona un conjunto de lineamientos que proponen la creación de representaciones, sin embargo, no garantiza la parte estética, pues este propósito, no está incluido dentro del enfoque de la investigación.

Por ultimo este paso se acopla a la fase de creación de cualquier metodología seleccionada en el paso anterior lo cual permite un desarrollo de software de una manera más eficaz y/o eficiente. Una vez se realice la creación del Plugin, el cual debe respetar el formato guía de lineamientos generales visto en la Ilustración 7 y las propiedades del Plugin consignadas en la Tabla 3, se procede al (**Paso 9**).

### **PASO 9: Testing del Plugin**

En el paso actual, se llevará a cabo una evaluación del Plugin creado, el cual puede ser soportado mediante la creación de pruebas de unidad, pruebas de integración, pruebas alfa y/o beta de las cuales se realiza una explicación más extensa al final de este paso. Finalmente se recomienda diligenciar una lista de chequeo con el objetivo de verificar el grado de personalización, ya sea de tamaño, color o textura, su reutilización en otros entornos y el cumplimiento de su funcionalidad principal que es representar un sensor, actuador o servicio.

Se tiene en cuenta que la metodología seleccionada en el **Paso 6** puede contener ya una fase de testing y los esquemas de las pruebas, pero debe entenderse que este paso solo proporciona recomendaciones, aunque si exige la aplicación de la lista de chequeo como adición para garantizar el grado de personalización y reutilización del Plugin, como se observa en el ejemplo de la Ilustración 11.

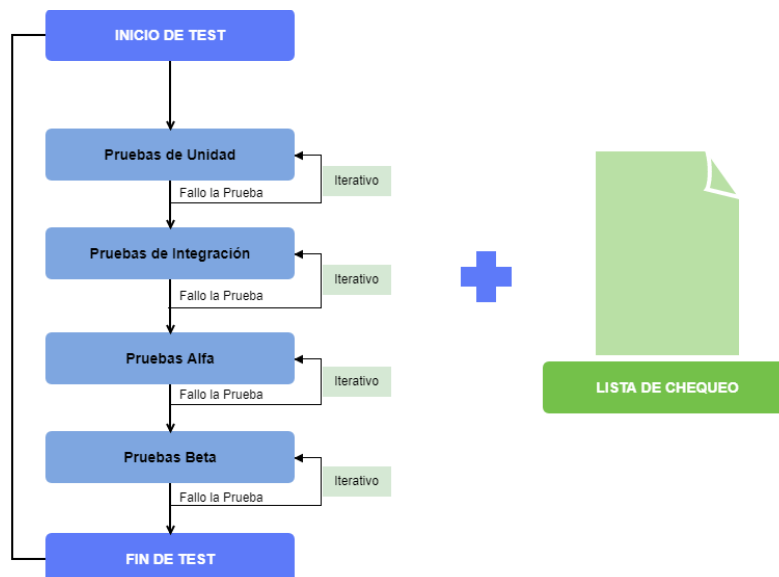


Ilustración 11 - Test de Plugin creado

A continuación, se presenta la lista de chequeo, que puede estar sujeta a modificaciones de acuerdo al grado de personalización o reutilización que el usuario desarrollador desee obtener.

### **LISTA DE CHEQUEO**



Una vez llevado a cabo el desarrollo de las pruebas pertenecientes a la metodología seleccionada, como pueden ser las sugeridas en el actual paso (pruebas de unidad, integración, alfa y beta), se recomienda como lineamiento lo siguiente:

**Lineamientos específico propuesto:**

- Implementar la lista de chequeo la cual va verificar si cumple con las características requeridas para los *Plugins* creados, en cuanto a personalización como es el caso de este trabajo de grado: cambio de textura o color o tamaño, también si es reutilizable en diferentes contextos y apto para ser usado por medio de realidad aumentada.

LISTA DE CHEQUEO PARA TEST DE PLUGIN						
Nombre del Plugin:			Fecha: DD/MM/AAAA			
Nº	Clasificación de características	Característica del Plugin	SI = 1	NO = 0	%	TOTAL
1	Personalización	¿El Plugin cumple con la característica que permite personalizar el tamaño?		X	8%	0*8% = 0
2		¿El Plugin cumple con la característica que permite personalizar el color?	X		8%	1*8% = 8%
3		¿El Plugin cumple con la característica que permite personalizar la textura?		X	8%	0*8% = 0%
4	Reutilización	El Plugin se puede reutilizar por medio de la exportación e importación de sus elementos.	X		30%	30%
5	Realidad Aumentada	¿La representación es interactiva en tiempo real?	X		18%	18%
6		<b>Para sensores:</b> ¿La representación es en 3D o en su defecto es un 2D contenido en un canvas o lienzo? <b>Para actuadores</b> ¿Su representación se presenta en un modelo 3D?	X		18%	18%
7	Pruebas Realizadas	¿El Plugin contiene pruebas de unidad, integración, alfa y	X		10%	10%



		beta, o pruebas definidas por la metodología seleccionada?				
<b>TOTALES</b>			<b>5</b>	<b>2</b>	<b>100%</b>	<b>84%</b>

Tabla 4 - Lista de chequeo para test del Plugin creado

Como podemos observar en la Tabla 4 el resultado de la prueba es dado por medio de porcentajes, esto para garantizar que el Plugin tenga las características deseadas por el modelo de referencia ARMoWoT. Se tiene en cuenta que el umbral de aceptación debe ser mayor o igual al **84%**, como se puede observar en el ejemplo anterior, el cual muestra un Plugin aceptado positivamente. El umbral del 84% se determina para garantizar las características básicas de ARMoWoT, por lo tanto se presentan los siguientes lineamientos:

**Sub-lineamientos específicos propuestos:**

- El Plugin debe contener las características de reutilización, ser una representación apta para mostrar mediante la realidad aumentada, haber realizado pruebas en el desarrollo y contener como mínimo una característica de personalización. Lo anterior en términos de la lista nos da el siguiente calculo:

$$30\% + 18\% + 18\% + 10\% + 8\% = 84\%$$

- También se aceptan *Plugins* que contengan las características de reutilización, ser aptas para la implementación en AR y contener las tres características de personalización, lo cual nos arroja el siguiente resultado:

$$30\% + 18\% + 18\% + 8\% + 8\% + 8\% = 90\%$$

Debido a lo anterior, lo que se busca es garantizar principalmente la reutilización, la personalización y el uso de AR para cada representación, evitando cualquier otra combinación no deseada de características. Por último, se presenta el plan de pruebas recomendado por el **Paso 9**:

**PRUEBAS UNITARIAS**

Las pruebas de unidad se realizan con el fin de comprobar un trozo de código desarrollado verificando que la realización del código esté funcionando según se espera, se recomienda continuamente estar realizando pruebas unitarias, permitiendo poco a poco ir fortaleciendo la creación del Plugin.

Las pruebas unitarias permitirán:

- Facilitar los cambios.
- Simplificar la integración.
- Errores más acotados y fáciles de localizar.

las pruebas de unidad aceleran el desarrollo óptimo aportando a obtener un código desacoplado debido a que cada función se piensa con el fin de que retorne un resultado y podrá ser testeado.

**PRUEBAS DE INTEGRACIÓN**

Las pruebas de integración se realizarán una vez aprobadas las pruebas de unidad, en las cuales se evalúan todos los elementos que componen un proceso, en este punto se realiza la prueba de integración con el objetivo de verificar que un conjunto de partes software creados funcionan acoplándolos.

### PRUEBAS ALFA

Una vez se generen pruebas de integración y finalmente se tenga una primera versión de la solución que representa el escenario planteado, se lleva a cabo la realización de una prueba alfa, la cual se describe como una prueba del Plugin por parte del cliente, en un ambiente controlado como observador el desarrollador registrando los errores y problemas de uso que se presenten.

El objetivo principal de llevar a cabo la prueba alfa, es verificar funcionalidad determinando si realmente las representaciones gráficas obedecen a los lineamientos propuestos en la presente investigación, adicionalmente se evalúa si el usuario que observa el escenario logra abstraer la información correcta a través de las representaciones.

Sin embargo, cabe resaltar que esta prueba pertenece a un producto aun inestable que todavía se encuentra en búsqueda de eliminación de errores. Sin embargo el *Plugin* satisface la mayoría de los requisitos que se plantean.

### PRUEBAS BETA

Las pruebas beta se refiere a la primera versión completa del producto desarrollado, el producto en este punto se considera un poco más estable y se crean con el objetivo de poner a prueba lo realizado, pero con un número determinado de usuarios para probarlo y de esta manera contrarlar la calidad y funcionalidad final.

### ZONA ITERABLE

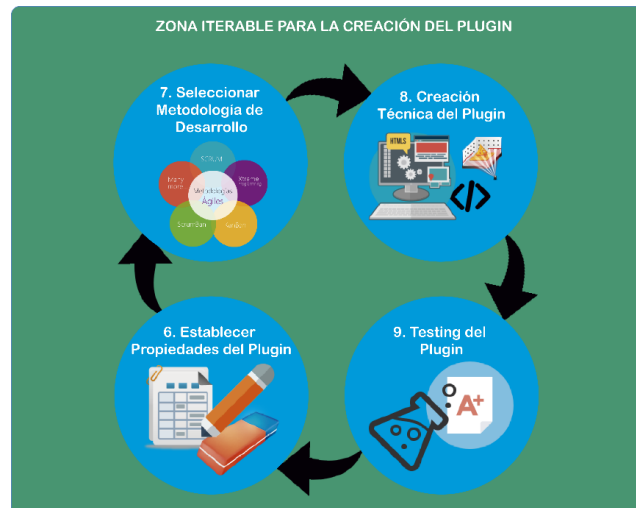


Ilustración 12 - Zona iterable para la creación del Plugin

Una vez realizada la fase de prueba denominada por la metodología seleccionada, como pueden ser las pruebas alfa, beta y siempre incluyendo la lista de chequeo se concluye todo el test del Plugin. Si se presenta el caso en el que el Plugin obtiene un puntaje menor al 84% en la lista de chequeo, se cataloga como un Plugin NO aceptable y se



procede a repetir los pasos (iniciando desde el **Paso 6**) iterativamente con el objetivo de ajustar las propiedades del *Plugin*, la metodología seleccionada y mejorar la creación del *Plugin*. Una vez se considere terminada una mejora al *Plugin*, se debe realizar nuevamente el test del *Plugin*, con el objetivo de que las condiciones NO aceptables sean superadas, ya sea que se iguale o se supere el umbral del 84%. En este punto, el test se considera como aprobado y el *Plugin* cumplirá con las características principales de ser personalizables y reutilizables en cualquier contexto de aplicación. Una vez el test del *Plugin* sea aprobado, se procede al (**Paso 10**) descrito a continuación:

### **PASO 10: Actualizar el repositorio con el nuevo Plugin creado**

Se procede a actualizar el repositorio de *Plugins* creados, y se ingresa el nuevo *Plugin* especificando las herramientas de realidad aumentada usadas, el entorno de desarrollo en la cual se creó y un tutorial de uso del *Plugin*.

Una vez ingresado los *Plugins* creados al repositorio, se procede al **Paso 3** para con el fin de verificar que todos los productos creados al aplicar el modelo se encuentran contenidos allí. En el caso de que faltara alguna representación se procede a todos los pasos anteriormente explicados. Pero si llegado el caso, después de la actualización del repositorio se tienen todos los *Plugins* necesarios para crear el proyecto AR, se procede simplemente al uso de ellos (**Paso 11**), reutilizando los *Plugins* en su escenario.

### **PASO 11. Usar Plugin en el nuevo proyecto AR**

Como último paso, se procede a descargar todos los *Plugin* necesarios para generar el proyecto en AR en IO para la WoT. Debido a lo anterior, se recomienda el siguiente lineamiento.

#### **Lineamientos específicos propuestos:**

- Escoger una metodología de prototipos, la cual como su nombre lo indica se enfoca en la creación de pequeños prototipos que cumplan con las funcionalidades que se requieren para el escenario propuesto, como por ejemplo el escenario de Domótica, casi siempre enfocado en desarrollos que usan realidad aumentada, obteniendo características como:
  - Mayor tolerancia a errores.
  - Priorización de requerimientos del desarrollador para su escenario propuesto.
  - Proceso evolutivo que crece a medida que sea desarrolla.

#### **3.2.2.2. Extensión del modelo**

- **Definición de un repositorio:** Un repositorio es un espacio digital centralizado donde se almacena, mantiene y comparte información de código software generalmente de acceso y uso público con el objetivo de tener un contenedor general de software subido en la web o mas conocido como internet.





- **Ventajas de uso de un repositorio:** En este punto se resalta lo beneficios y ventajas por las cuales se escogió el uso de un repositorio para los *Plugins* desarrollados en base al modelo de referencia propuesto anteriormente. Dentro de los beneficios se pueden determinar los siguientes ítems:
  - **Preservación a largo plazo:** El repositorio de *Plugins* propuesto, aporta una preservación de los datos (*Plugins* o prefabricados) almacenados en él, ofreciendo una mayor fiabilidad de las soluciones, inclusive más que cualquier otro centro de almacenamiento virtual como lo que ofrecen las páginas web personales, los servidores de revistas virtuales entre otras.
  - **Comunicación:** La comunicación que ofrece un repositorio es de vital importancia, ya que dota a cualquier persona de permisos, para que tenga acceso libre a un intercambio de información ya sea científica o de cualquier otro tipo.
  - **Evaluación:** Se habla de evaluación debido a que esta es la que refleja la actividad intelectual de la presente investigación, permitiendo evaluar los productos y resultados en un solo lugar virtual.
  - **Unidad:** Todos los *Plugins* desarrollados se encontrarán almacenados en una misma base de datos permitiendo así su fácil recuperación o acceso.
  - **Ubicación del repositorio de Plugins:** El repositorio de *Plugins* será encontrado en la siguiente dirección:  
<https://bitbucket.org/account/user/ariotteam/projects/ARMOWOT>.
- **Tutorial para el uso de Plugins:** Para complementar los *Plugins*, se crea una página web donde se deposita la información principal del modelo ARMoWoT y los tutoriales explicativos para reutilizar y personalizar los *Plugins* en otros proyectos para IO con la técnica de AR, a continuación, se presenta la URL de la página web:

<https://armowot.000webhostapp.com/>

### 3.2.3. Fase 3: Diseño de prueba preliminar del modelo de referencia

El objetivo principal de la prueba preliminar, es comprobar si la solución propuesta del modelo de referencia junto a sus características internas como: lineamientos, pautas, recomendaciones y buenas prácticas de desarrollo, otorgan a los desarrolladores la facilidad de crear soluciones óptimas personalizables y reutilizables en cualquier contexto de la WoT. De igual manera la prueba genera una retroalimentación del modelo, fortaleciéndolo en caso de encontrar alguna falencia y a la vez permite observar los beneficios que concede a los usuarios desarrolladores, medidas en el ahorro de tiempo al crear modelos 2D o 3D que representen actuadores, sensores y servicios de los IO.





### 3.2.3.1. Desarrollo de la prueba preliminar del modelo de referencia.

#### **PASO 1: Análisis y diseño del escenario**

Con base en los lineamientos del modelo de referencia, se plantean un escenario que permita representar gráficamente los datos de los sensores y servicios de un producto inteligente con sensores de nivel y temperatura de líquido. Los servicios que se quieren representar gráficamente son los que otorgan los sensores de temperatura de líquido DS18B20, un sensor de nivel de líquido y un actuador led RGB, notificando al usuario los datos que arrojan mediante representaciones gráficas, para que soporte la toma de decisiones.

#### **PRODUCTO INTELIGENTE → Termo Inteligente:**

Se desea crear un producto inteligente denominado Termo Inteligente, el cual permita notificar mediante realidad aumentada la temperatura y el nivel de líquido que contiene, con el fin de soportar la toma de decisiones que desee el usuario del producto.

El producto inteligente deriva tres casos de los cuales se describen a continuación:

**Caso 1:** El termo mediante realidad aumentada realiza una representación gráfica de los datos que arrojan los sensores, notificando al usuario la temperatura en la que se encuentra el líquido y el nivel del líquido. Si el líquido contenido en el termo está hirviendo o de alguna manera se encuentra muy caliente, el producto notifica mediante realidad aumentada a la representación en 2D o 3D, la temperatura en la que se encuentra el líquido, con el objetivo de que la persona no se lastime a la hora de consumirlo.

**Caso 2:** En el caso de que el líquido se encuentre a temperatura ambiente, el producto notifica al usuario con una representación gráfica adecuada, aportando una posibilidad de mejor abstracción de la información y soporte en la toma de decisión con mayor agilidad. La representación debe indicar al usuario que el líquido está disponible para el consumo sin correr algún tipo de riesgo.

**Caso 3:** El último caso que se puede presentar, es cuando los datos otorgados por los sensores representen el nivel del líquido contenido en el recipiente y la temperatura baja, indicando que el líquido se encuentra frío o muy frío para que el usuario tome una decisión.

#### **PASO 2: Investigar Herramienta Adecuada**

Basados en los lineamientos que componen al modelo de referencia y en el análisis del escenario que se propone desarrollar, se procede a realizar el paso numero dos que describe llevar a cabo el estudio técnico de las herramientas para el desarrollo de los modelos, el presente paso se compone de las siguientes tareas:

- Selección del entorno de desarrollo (2.A).



- Selección de las herramientas que nos permitan crear los modelos de representación gráfica mediante *Plugins* en la técnica de realidad aumentada (2.B).
- Definición del número de componentes entre sensores, actuadores y servicios de los diferentes objetos inteligentes que hacen parte del escenario planteado, indicando el número de *Plugins* a implementar.
- Como paso final, realizar una consulta de la información que está contenida en el formato guía de lineamientos generales para representaciones gráficas de los IO de acuerdo a las características de sensores, actuadores y servicios que los componen.

## PASO 2.A.

### **Criterios de selección para el entorno de desarrollo:**

Una vez observadas las características del entorno de desarrollo descritas en el ANEXO B, se selecciona Unity3D debido a que, es uno de los entornos actualmente más usados para la creación de proyectos con realidad aumentada. Lo anterior se soporta con base en los trabajos relacionados indicados en el marco teórico de la presente investigación, contando con los beneficios de ser un entorno multiplataforma, con un total de 27 plataformas exportables entre las cuales se destacan (iOS, Android, Windows Phone, Mac, Linux, Samsung Smart tv, Microsoft Hololens, Xbox One, entre otros). Además posee un gran soporte para la conexión a la IoT, tolerando protocolos de comunicación IP, que se requieren para llevar a cabo la conexión con dispositivos hardware, permitiendo un acople factible con la tecnología de realidad aumentada.

## PASO 2.B.

### **Criterios de selección para la herramienta software de realidad aumentada:**

Una vez observadas las herramientas descritas en el ANEXO B, se selecciona el kit de desarrollo SDK<sup>15</sup> Vuforia debido a que, la herramienta es de tipo libre con opción de una SDK comercial, aplicable o compatibles con entornos como iOS, Unity3D, Windows Mobile entre otros. Adicionalmente posee un API de contenido siendo todas estas características muy importantes a la hora de crear una aplicación basada en realidad aumentada.

## PASO 2.C.

Una vez seleccionado la herramienta de realidad aumentada Vuforia y el entorno de desarrollo Unity3D:

El número de *Plugins* (prefabricado) a representar gráficamente para el producto del Termo Inteligente teniendo en cuenta la cantidad de componentes como sensores, actuadores y servicios son:

- Sensor de Temperatura = Plugin 2D Temperatura + Plugin 3D Temperatura = **2**.
- Sensor de Nivel = Plugin 2D Nivel + Plugin 3D Temperatura = **2**.
- Actuator Led = Plugin 3D LedEncendido = **1**.

<sup>15</sup> SDK – software development kit, conjuntos de herramientas y librerías de código abierto (Open source) para soporte en el desarrollo de software que le permite al desarrollador la creación de aplicaciones para un sistema concreto



Por lo tanto, se tiene 5 posibles representaciones para mostrar los datos y metadatos del IO Termo Inteligente. Debido a el entorno de desarrollo, se debe saber que las representaciones de este tipo, son llamadas Prefab y no *Plugins*, pero para esta tesis se mantendrá el termino *Plugins*, para evitar tener que cambiar de términos al momento de abordar un entorno de desarrollo. Se requiere que cada Plugin se pueda comunicar mediante protocolos de comunicación como MQTT o API REST y sus animaciones representen debidamente los datos obtenidos de los IO. Cuando se trata de un modelo en 3D, en el modelo grafico o su apariencia pueden cambiar dependientemente del dato obtenido que arrojan los sensores.

### **PASO 3: Examinar el Repositorio de Plugins Creados**

Se examina el repositorio de Plugin creados, con el objetivo de encontrar si ya se han creado los *Plugins* que representen gráficamente los dispositivos que se necesita para el escenario propuesto de los dos productos inteligentes. Sin embargo, no se encuentra ningún Plugin que se acople a los datos del sensor de temperatura, nivel de un líquido y un actuador led RGB, por tal motivo se procede al siguiente paso.

### **PASO 4: Revisar Formato Guía de Lineamientos para Representaciones Graficas de IO**

En el presente paso se realiza la revisión del formato guía para crear representaciones gráficas de los IO de acuerdo a las características de sensores, actuadores y servicios que los componen presente en la Ilustración 7, y se observan los lineamientos que se proponen para los sensores de temperatura de líquido DS18B20, el sensor de nivel de líquido y el actuador led RGB, siendo estos el total de los componentes de los productos inteligentes necesarios para representar el escenario propuesto. Como se da el caso en el que, en la Ilustración 7 se encontraron todos los componentes necesarios con sus respectivos lineamientos, entonces se procede a realizar el (Paso 6).

### **PASO 5: Retroalimentar Formato Guía de Lineamientos para Representaciones Graficas de IO**

El presente paso no se realizó debido a que si encontró la información que se requiere de los sensores, actuadores y servicios en el formato guía contenido en la Ilustración 7 y presentado en su extensión en el ANEXO C, por tal motivo se procede a realizar el siguiente paso.

### **PASO 6: Establecer Propiedades del Plugin**

En el presente paso se procede a establecer las propiedades o metadatos del Plugin, los cuales describen como guía en la Tabla 5 compuesto por el nombre del Plugin, una breve descripción del problema a solucionar y la descripción de la solución que se plantea realizar:

	<b>Propiedades Del Plugin</b>
--	-------------------------------



Metadatos de un Plugin		Nombre de los Plugins	Descripción del problema a solucionar	Descripción general de la posible solución
<b>PLUGINS</b>	Plugin que representa el servicio de temperatura	Plugin 2D Temperatura  Plugin 3D Temperatura Textura	Se requiere crear un modelo que represente los servicios del sensor ds18b20 el cual mide la temperatura de un líquido.	Se selecciona el entorno de desarrollo Unity3D con base en el estudio de las características y buenos aportes a la hora de crear un modelo que pueda ser reutilizable y represente gráficamente el servicio otorgado por el módulo de nivel y el sensor ds18b20. Estos sensores arrojan datos mediante una conexión MQTT los datos son entregados al motor de desarrollo Unity3D, una vez el motor obtenga los datos de los sensores, mediante librerías procesa los datos, permitiendo a los desarrolladores generar el Prefabricado (Plugin). Como resultado final se obtienen los <i>Plugins</i> creados y mediante AR se logra observar la representación gráfica de los servicios que otorgan el objeto inteligente.
	Plugin que representa el servicio del actuador Led RGB	Plugin 3D ActuadorSeñal	Se requiere crear un modelo que represente los servicios del actuador, representado físicamente por un led RGB.	
	Plugin que representa el servicio de nivel	Plugin 2D Nivel  Plugin 3D Nivel	Se requiere crear un modelo que represente los servicios del módulo de nivel para Arduino el cual mide el nivel de un líquido.	

Tabla 5 - Metadatos de los Plugins pertenecientes a la prueba preliminar del modelo de referencia ARMoWoT.

Una vez establecido los metadatos que recomienda el modelo de referencia, se procede al siguiente paso.

### **PASO 7: Seleccionar Metodología de Desarrollo**

En el presente paso se lleva a cabo la selección de la metodología de desarrollo que se adecue a las necesidades. Por tal motivo se siguen las recomendaciones del modelo de referencia ARMoWoT y se decide aplicar la metodología ágil de desarrollo llamada XP en el cual se acogen las fases de planeación, codificación y pruebas, donde se realizara solo una iteración debido a que se trata de una prueba de concepto del modelo, por lo tanto, la magnitud del desarrollo es corto. Por último, se acoge el desarrollo de software por parejas llevando un manejo colectivo del código.

### **PASO 8: Creación Técnica del Plugin**

**Planeación:**



Las historias de usuario<sup>16</sup> se documentan en el ANEXO D, por lo que se hace una pequeña planeación para la creación técnica de los *Plugins*, que representaran los datos del objeto inteligente llamado Termo Inteligente.

**Codificación:** Creación técnica del *Plugin* (Prefab en unity3d):

En este punto se realiza la creación técnica de los *Plugins*, teniendo en cuenta los lineamientos anteriormente documentados y recomendados por el modelo de referencia ARMoWoT. Los códigos fuentes, documentación y otros recursos para el desarrollo de los *Plugin* se encuentran en el ANEXO D de la presente investigación, sin embargo, se presenta información en imágenes del resultado final del desarrollo y una descripción.

- **Lógica de Comunicación:** Para la comunicación por medio de los protocolos MQTT y API REST, se desarrollan las clases llamadas **“LogicCommunication”** y **“EditorCommunication”**, las cuales son dependientes una de otra. Su propósito es proporcionar un medio de comunicación entre los IO físicos y las representaciones graficas 2D y 3D. Para generar la comunicación por MQTT se utiliza una librería llamada m2mqtt4unity<sup>17</sup>, quien proporciona la lógica para ese protocolo de comunicación. La lógica es usada en cada *Plugin*, representado por medio de un ítem llamado **“Comunicación”**. Como se puede observar en la Ilustración 13, también se ve la ventana de edición para seleccionar entre los protocolos de comunicación mencionados al principio de este párrafo. El modo de uso del código se encuentra consignado en el ANEXO D.

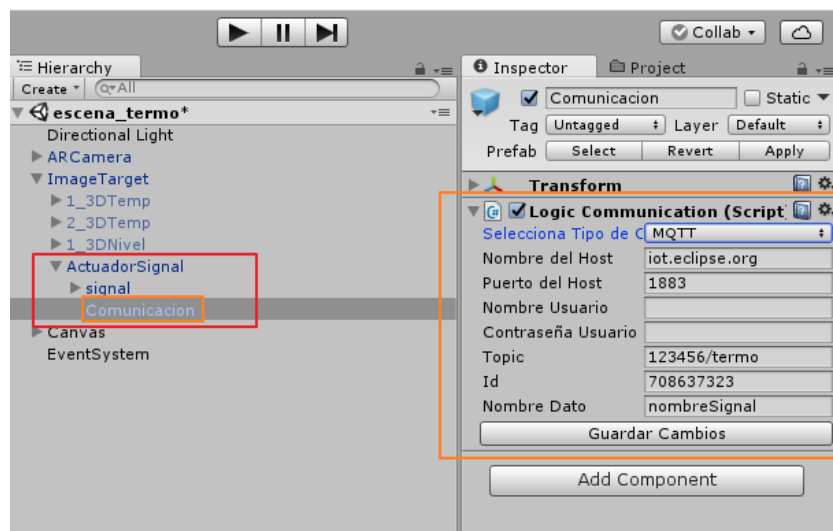


Ilustración 13 - Ejemplo de Comunicación en Unity 3D

- **Detección AR Canvas:** Se desarrolla una clase llamada **“DetectionAR”** la cual permite que un elemento Canvas<sup>18</sup> pueda ser marcado como un elemento AR sin

<sup>16</sup> Se describen como la representación o especificación de un requisito escrito, utilizando el lenguaje común de usuario.

<sup>17</sup> <https://archive.codeplex.com/?p=m2mqtt4unity>

<sup>18</sup> El Canvas es el área donde todos los elementos de tipo 2D deben estar, en el entorno de desarrollo de Unity 3D. <https://docs.unity3d.com/es/current/Manual/UICanvas.html>

necesidad de ser un hijo de los Target<sup>19</sup> conocidos en Vuforia. La clase solo se puede usar en el entorno de desarrollo de Unity por cualquier Plugin de tipo 2D. El código tiene la función de permitir agregar de 1 a N elementos, para ser activados al momento de detectar el identificador, ya sea un código QR u otro tipo de identificador. Esto si se desea independencia entre los identificadores y las representaciones, como a veces sucede con los elementos que componen al Canvas.

Los anteriores códigos descritos, son fundamentales a lo largo de la construcción de cada *Plugins*, debido a que la lógica de la comunicación es igual para todos y la detección AR Canvas es usada en todos los *Plugins* de tipo 2D. A continuación se presentan dos ejemplos de los cinco *Plugins* creados en total, la descripción de cada *Plugins* se encuentra contenida en el ANEXO D.

- **Plugin 2D Temperatura:**

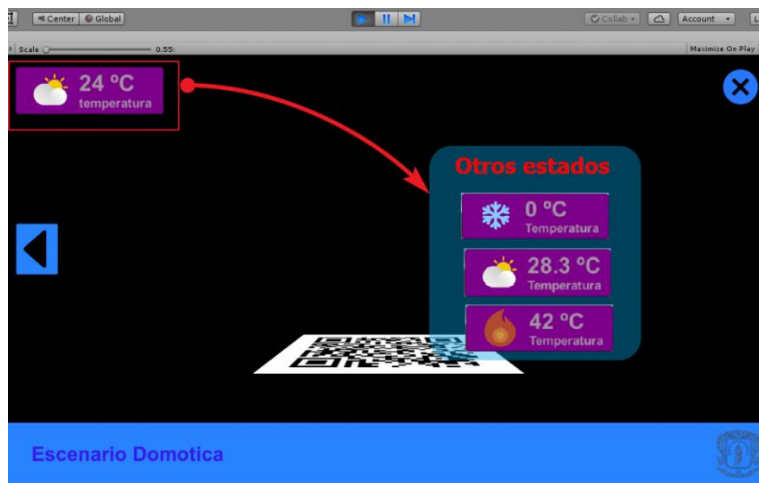


Ilustración 14 - Plugin 2D Temperatura

<sup>19</sup> Es el nombre por el que se conocen los diferentes tipos de identificadores para Realidad Aumentada, se puede obtener más información en el marco teórico de [16] O. E. Guzmán-Álvarez, "Interfaz de realidad aumentada para la internet de las cosas," Departamento de Sistemas, Universidad del Cauca, 2016.

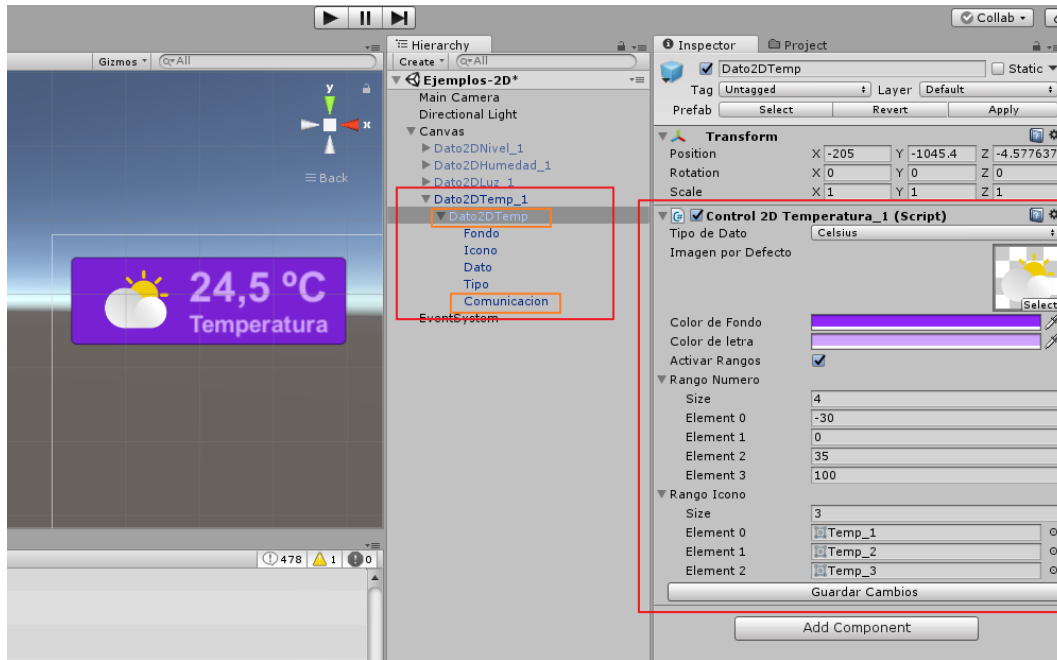


Ilustración 15 - Plugin 2D Temperatura y su ventana de personalización

Como se puede observar el propósito de este prefabricado es representar los datos del sensor de Temperatura en una interfaz 2D que pueda ser manipulada en un Canvas para AR. Para generar una ventana de personalización se desarrolló dos clases. Las dos primeras llamadas **“Control2DTemperatura\_1”** y **“Editor2DTemperatua\_1”** son dependientes una de la otra, esto debido a que la primera proporciona la lógica para cambiar los iconos(Textura) dependiendo de los cambios de temperatura y la segunda proporciona la funcionalidad para personalizar el Plugin en tiempo real, antes de iniciar la compilación en el entorno de desarrollo.

- **Plugin 3D Temperatura Textura:**



Ilustración 16 - Plugin 3D Temperatura para Termo Inteligente



Como se puede observar en la Ilustración 16 el Plugin refleja la temperatura por medio de un modelo 3D, el cual debe de componerse de un solo material<sup>20</sup>, para así mostrar texturas como el hielo o la lava, también se puede escalar a un tamaño adecuado. La personalización del modelo se da por medio de las clases “**Control3DTemperatura\_1**” y “**Editor3DTemperatura\_1**” para generar una ventana de personalización.

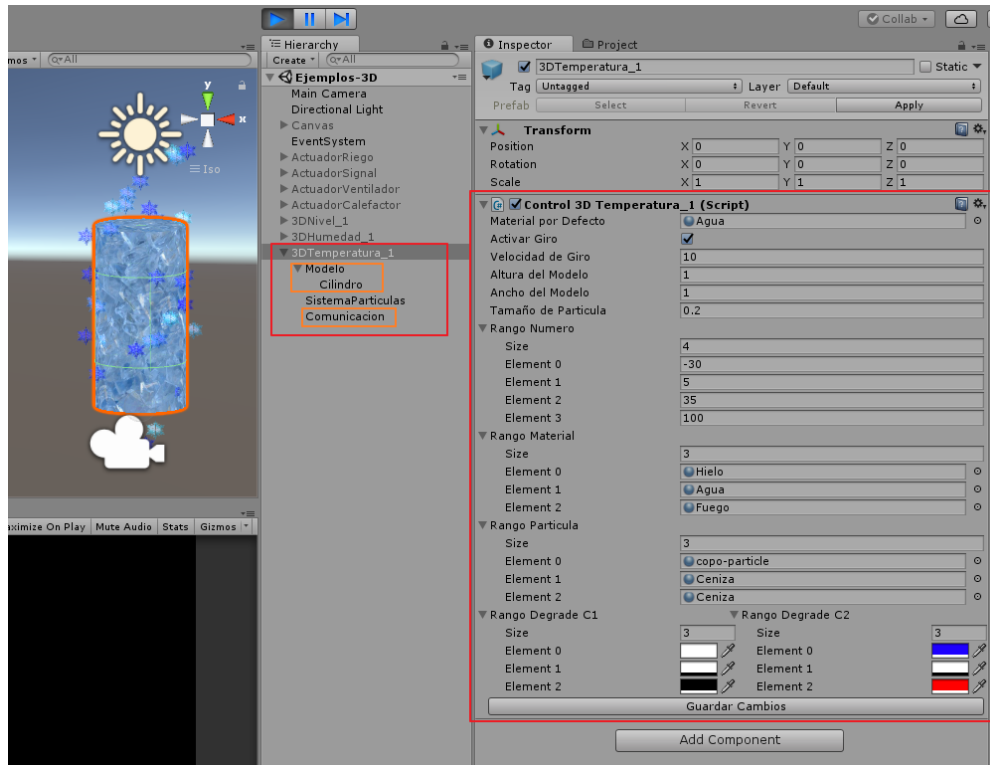


Ilustración 17 - Plugin 3D Temperatura y su ventana de personalización para Termo Inteligente

## **PASO 9: Testing del Plugin**

En el paso actual, se lleva a cabo un testeo de los Plugin creados, denominados como Plugin por el entorno de desarrollo. El test se soportará en las pruebas que se proponen en el modelo, sin embargo se resalta que durante el desarrollo del modelo se llevaron a cabo pruebas de unidad, integración y pruebas alfa y beta para obtener finalmente el prefabricado creado y lo más óptimo posible. Posteriormente se realiza la lista de chequeo, donde se determinará si permite realizar la personalización en cuanto a tamaño, color o textura, de igual manera la reutilización y si se llevaron a cabo las pruebas anteriormente mencionadas.

### **LISTA DE CHEQUEO:**

Todas las listas de chequeo se encuentran consignadas en el ANEXO D, por lo tanto aquí se presentan los siguientes resultados:

<sup>20</sup> Textura con la que se quiere mostrar un modelo 3D en el entorno de Unity, puede ser vidrio, tierra, madera etc. Más información en: <https://docs.unity3d.com/es/current/Manual/class-Material.html>





- Plugin 2D Temperatura: **92%**.
- Plugin 3D Temperatura Textura: **100%**.
- Plugin 2D Nivel: **92%**.
- Plugin 3D Nivel: **100%**.
- Plugin 2D Actuator Señal: **84%**.

Por los resultados anteriores, obtenidos de la lista de chequeo, se puede verificar que los *Plugins* creados para representar gráficamente el termo inteligente, así como sus sensores de nivel y temperatura, estuvieron igual o por encima del 84%. Lo cual cumplen con las expectativas iniciales que se buscan en prefabricados realizados en base al modelo ARMoWoT.

### **PASO 10: Actualizar el Repositorio con los Nuevo Plugins Creados**

Se procede a agregar los *Plugins* nuevos en el repositorio, ubicado en:

<https://bitbucket.org/account/user/ariotteam/projects/ARMOWOT>

A su vez se agregan nuevos tutoriales, para conocer el manejo de los *Plugins*:

<https://armowot.000webhostapp.com/documentacion/>

Al incluir los nuevos *Plugins* (Prefabs) creados, se debe indicar la tecnología utilizada con la cual se llevó a cabo la implementación. Para este caso en la prueba preliminar del modelo de referencia se crearon los *Plugins* en las herramientas de realidad aumentada VuforiaSDK y el entorno de desarrollo Unity3D.

Una vez actualizado el repositorio, se procede a revisar nuevamente el repositorio con el fin de verificar si quedaron correctamente subidos, por último, se seleccionan y descargan para iniciar los nuevos proyectos.

### **PASO 11: Usar Plugin en el Nuevo Proyecto AR**

El presente paso, se trata del uso de los *Plugins* en diferentes contextos, según sea la necesidad. La reutilización de los *Plugins* será confirmada en este punto, permitiendo que cada usuario de Plugin logre perfectamente la personalización y reutilización en cualquier contexto de aplicación.

A continuación, se tienen las imágenes captadas de las representaciones gráficas del sensor de temperatura DS18B20, el sensor de nivel y el actuador led RGB:

- **Termo Inteligente:** En este punto se prueban los *Plugins* desarrollados para la prueba de concepto compuesta por los sensores de nivel, temperatura del líquido y un actuador Led para el termo, posteriormente se aplica la técnica de realidad aumentada (La técnica es mostrada en los ANEXOS D) se capta el IO y dependiendo de los datos que arrojen los sensores y actuadores, se muestra la siguiente interfaz en AR para el Termo Inteligente.

## Termo Inteligente

En el siguiente desarrollo se puede observar la prueba de concepto denominado Termo Inteligente. Donde se utiliza ARMoWoT (Modelo de Realidad Aumentada para la Web de las Cosas) para generar la representación de cada sensor y actuador que componen el Objeto Inteligente.



Iniciar



Yuri Viviana Daza Gaviria - Juan Felipe Valencia Mosquera

*Ilustración 18 - Escena inicial de Termo Inteligente*



Termo Inteligente



*Ilustración 19 - Representación gráfica en AR para el IO Termo Inteligente*



Ilustración 20- Termo Inteligente con diferentes temperaturas



## CAPÍTULO 4 CONSTRUCCIÓN DEL ESTUDIO DE CASO

Durante este capítulo, se lleva a cabo la construcción del estudio de caso, el cual permitirá testear la funcionalidad del modelo de referencia ARMoWoT propuesto en el anterior capítulo, centrándose en establecer parámetros de personalización y reutilización de los *Plugins* creados, independientemente del contexto en el que se encuentre el IO.

El motivo de realizar un solo estudio de caso, se debe a que con un solo proyecto de interfaz AR que represente gráficamente todos los IO de un escenario, se tendrán los primeros *Plugins* para el repositorio. Una vez se almacene y se documenten los *Plugins* en el repositorio, pueden ser reutilizados en una evaluación en otros proyectos con diferentes contextos, así se evaluará en su totalidad el modelo de referencia.

Para la creación del Proyecto AR del escenario, se tiene en cuenta la metodología de Guzmán-Álvarez [16] llamado "*Enfoque metodológico para la creación de interfaces de AR para la IoT*", la cual propone las siguientes cuatro fases: fase de análisis, fase de diseño, fase de construcción y fase de validación además de adaptar los elementos sugeridos para la creación de prototipos funcionales [41]. Por último, se aclara que para este estudio de caso se van a realizar todos los pasos propuestos en ARMoWoT, pero se debe tener en cuenta que es un modelo de referencia y no es necesario seguir cada paso de la consigna, solo para su verificación se realizan en este proyecto todos los pasos sugeridos.

### 4.1. IMPLEMENTACIÓN DEL MODELO DE REFERENCIA ARMoWoT

#### 4.1.1. Paso 1: Análisis y diseño del escenario

El escenario de aplicación en este proyecto será el de Domótica, teniendo en cuenta que la domótica se conforma por todos los elementos contenidos en una edificación, en este caso una oficina. El requerimiento más relevante es generar una representación por medio de la técnica de AR para los siguientes objetos inteligentes: regulador de humedad, regulador de temperatura y regulador de luz. El producto final que se desea obtener es una aplicación para dispositivo móvil Android, donde pueda observar en una representación 3D los datos de mis IO y en 2D los datos planos.

#### 4.1.2. Paso 2: Investigar herramienta adecuada

- **Tarea A:** Para seleccionar el entorno de desarrollo se tiene en cuenta la tabla contenida sobre entornos de desarrollo en el ANEXO B y tres ventajas fundamentales como son el número de plataformas exportables, el soporte con AR y la baja curva de aprendizaje, la elección para este proyecto es Unity 3D.
- **Tarea B:** Tomando en cuenta que el entorno de desarrollo es Unity 3D y tomando como base la tabla sobre herramientas de realidad aumentada contenida en el





ANEXO B, se selecciona el SDK de Vuforia, debido a que la última versión de Unity viene integrado con Vuforia.

- **Tarea C:** Un total de 10 *Plugins* o Prefab (Debido al entorno de desarrollo Unity)
  - Regulador de Temperatura = 1 Actuador de Calefacción + 1 Actuador Ventilador + 1 Sensor de Temperatura = Plugin 3D Calefacción + Plugin 3D Ventilador + Plugin 2D Temperatura + Plugin 3D Temperatura = **4**
  - Regulador de Humedad = 1 Actuador de Riego + 1 Sensor de Humedad = Plugin 3D Riego + Plugin 2D Humedad + Plugin 3D Humedad = **3**
  - Regulador de Luz = 1 Actuador Bombilla + 1 Sensor de Luz = Plugin 3D ActuadorLuz + Plugin 2D Luz + Plugin 3D Luz = **3**
- **Tarea D:** Como no existen lineamientos significativos, se procede al siguiente paso.

#### 4.1.3. Paso 3: Examinar el repositorio de Plugins creados

Se procede a realizar una búsqueda de las representaciones deseadas, teniendo en cuenta el entorno de desarrollo y la herramienta de AR, en los siguientes enlaces:

- Documentación de los *Plugins*:  
<https://armowot.000webhostapp.com/>
- Repositorio de *Plugins* para Unity y SDK de Vuforia:  
<https://bitbucket.org/account/user/ariotteam/projects/ARMOWOT>

Después de realizar la búsqueda, se observa que existe una representación para el Sensor de Temperatura y otro en forma de señal activa o inactiva que funciona muy bien para el Actuador Bombilla. Pero al observar más de cerca la representación del Sensor de Temperatura no se acopla a lo deseado y no se encuentran más representaciones que se puedan usar en el proyecto. Por lo tanto se procede a realizar el Paso 4.

#### 4.1.4. Paso 4: Revisar formato guía de lineamientos generales para representaciones graficas de IO

Al revisar el formato guía contendió en el ANEXO C y explicado por medio de una plantilla en Ilustración 7 . Se encuentran varias consignas que orientan a desarrollar un Plugin óptimo para el Sensor de Temperatura y el Actuador Bombilla. Pero no se encuentran lineamientos para generar representaciones en los demás sensores y actuadores que corresponden a los IO del escenario de Domótica. Por lo tanto, se procede a realizar el Paso 5.

#### 4.1.5. Paso 5: Retroalimentar formato guía de lineamientos para representaciones graficas de IO.

Para retroalimentar el formato guía, se ejecuta una aplicación de escritorio desarrollada por los creadores del modelo, como un primer prototipo de búsqueda rápida. En donde se solicitan los parámetros correspondientes a la Ilustración 7. Por último, se agrega la información en el ANEXO C, dando por cumplido el paso 5, ya con esa información se procede a realizar el paso 6.



#### 4.1.6. Paso 6: Establecer propiedades del Plugin

Debido a los pasos anteriores, ahora se tiene noción de los componentes que deben contener los *Plugins* y la importancia de ser construidos con base a la reutilización y personalización de la representación, ya sea tamaño, color o textura. Por lo tanto se procede a llenar la Tabla 3, en la cual se tendrá en cuenta en todo el proceso de desarrollo de los *Plugins*. Ahora bien recordando la Tabla 1 y la Tabla 2, donde se hablan acerca de la terminología cambiante con respecto al entorno de desarrollo, por lo tanto el producto resultante debido al entorno de desarrollo se llama Prefab pero para el estudio de caso se mantendrá el término de Plugin. Teniendo en cuenta lo anterior se procede a llenar la Tabla 3, de los *Plugins* faltantes para realizar el proyecto en AR para el escenario de Domótica:

Metadatos de un Plugin		Propiedades Del Plugin		
		Nombre de los Plugins	Descripción del problema a solucionar	Descripción general de la posible solución
<b>PLUGINS</b>	Plugin que representa el dispositivo de Calefacción	Plugin 3D Actuador Calefacción	Se desea obtener una funcionalidad adicional que represente la actividad e inactividad de un calefactor por medio de una representación 3D.	Se realizaran prefabricados 2D/3D que permitan visualizar mediante AR los datos que arrojan los diferentes componentes de los IO como el calefactor, el ventilador y sensor de temperatura ambiente, con el objetivo de mostrar gráficamente todos los datos que arrojan.
	Plugin que representa el dispositivo de Ventilador	Plugin 3D Actuador Ventilador	Se desea obtener una funcionalidad adicional que represente la actividad e inactividad de un ventilador por medio de una representación 3D.	
	Plugin que representa el sensor de temperatura ambiente – LM35	Plugin 3D Volcán temperatura	Se requiere crear un modelo que represente los servicios del LM35 el cual mide la temperatura ambiente.	



<p><b>Plugin que representa el dispositivo de Riego</b></p>	<p>Plugin 3D Actuador de Riego</p>	<p>Se requiere representar la actividad e inactividad de un dispositivo de riego por medio de 3D.</p>
<p><b>Plugin que representa el sensor de Humedad de tierra – Moinsture Sensor v1.4</b></p>	<p>Plugin 2D Humedad  Plugin 3D Humedad</p>	<p>Se requiere crear una interfaz 2D y un modelo 3D que represente los servicios del Moinsture Sensor v1.4 el cual mide la humedad en la tierra.</p>
<p><b>Plugin que representa el sensor de Luz – TSL2561</b></p>	<p>Plugin 2D Luz  Plugin 3D Luz</p>	<p>Se requiere crear una interfaz 2D y un modelo 3D que represente los servicios de TSL2561 el cual mide la iluminación en el ambiente.</p>

Tabla 6 - Metadatos de los Plugins pertenecientes al escenario de Domótica del modelo de referencia ARMoWoT.

#### 4.1.7. Paso 7: Seleccionar metodología de desarrollo

En este paso se va a seleccionar una metodología ágil, esto debido a que no se va desarrollar un producto total sino micro desarrollos para la construcción de cada representación faltante para el proyecto AR de Domótica. Por lo tanto se elige la metodología XP [42] por su adaptabilidad y flexibilidad, la cual nos proporciona fases que se acoplan perfectamente a los pasos 8 y 9. Para iniciar la metodología se procede a realizar la planificación que se seguirá para desarrollar las representaciones:

##### Planificación inicial:

En esta planificación inicial se contempla documentar las historias de usuario necesarias, que corresponden a las representaciones que se desean generar en el **Paso 8**. Teniendo





en cuenta la Tabla 6 y el formato guía de lineamientos generales, los cuales proporcionan una noción más clara de los componentes personalizables a desarrollar en las representaciones. En esta primera planificación resulto la redacción de las historias de usuario, las cuales a medida que se conocían más aspectos de las representaciones fueron cambiando hasta concluir con lo que podemos observar en el **Paso 8**.

#### **4.1.8. Paso 8: Creación técnica del Plugin**

Como las historias de usuario son varias y lo que se quiere resaltar en este paso es el desarrollo de los *Plugins*, se consignan todas las historias de usuario en el ANEXO D, al igual que código fuente y listas de chequeo.

**Iteración 1:** Historias de Usuario realizadas en esta iteración: H1, H2 y H3.

Se debe tener en cuenta que todos los *Plugins* desarrollados en esta tesis tienen igual lógica para realizar la comunicación, se puede configurar como se muestra en la Ilustración 13, para obtener más información existe más documentación en la siguiente página web, la cual fue desarrollada como un artefacto más de la presente tesis: <https://armowot.000webhostapp.com/>.

Con el fin de simplificar la codificación de cada *Plugins* se alberga información técnica, imágenes resultantes del desarrollo y otros artefactos, en el ANEXO E para los siguientes *Plugins*:

- Plugin 3D Actuador Calefacción.
- Plugin 3D Actuador Ventilador.
- Plugin 3D Volcán Temperatura.

**Iteración 2:** Historias de Usuario realizadas en esta iteración: H4, H5 y H6.

En la iteración dos del desarrollo, se construye en su totalidad los siguientes *Plugins* (tener en cuenta que la información extra se contiene en el ANEXO E):

- Plugin 3D Actuador Riego.
- Plugin 2D Humedad.
- Plugin 3D Humedad.

**Iteración 3:** Historias de Usuario realizadas en esta iteración: H7 y H8.

En esta iteración se realiza la construcción técnica de los siguientes *Plugins*:

- Plugin 2D Luz.
- Plugin 3D Luz.



#### 4.1.9. Paso 9: Testing del Plugin

Se lleva a cabo un testeo de los prefabricados creados por cada uno de los objetos inteligentes del escenario representado, el cual se soportará en las pruebas que se proponen en el modelo, sin embargo se resalta que durante el desarrollo del modelo se llevaron a cabo pruebas de unidad, integración y pruebas alfa y beta para obtener finalmente el prefabricado creado y lo más óptimo posible.

Posteriormente se realiza la lista de chequeo, donde se determinará si permite realizar la personalización en cuanto a tamaño, color o textura, de igual manera la reutilización y si se llevaron a cabo las pruebas anteriormente mencionadas. Debido a que todos los cuadros de la lista de chequeo se encuentran en el ANEXO E, se procede a mostrar los resultados de los test por cada Plugin:

- Plugin 3D Actuador Calefacción: **84%**.
- Plugin 3D Actuador Ventilador: **84%**.
- Plugin 3D Volcán temperatura: **84%**.
- Plugin 3D Actuador de Riego: **84%**.
- Plugin 2D Humedad: **92%**.
- Plugin 3D Humedad: **92%**.
- Plugin 2D Luz: **92%**.
- Plugin 3D Luz: **84%**.

Los anteriores resultados de las listas de chequeo permiten verificar que los *Plugins* creados para representar gráficamente el escenario propuesto en el estudio de caso, cumplen con las expectativas, debido a que todos están en un rango igual o mayor al 84% de aceptación, este es el resultado que se buscan al aplicar el modelo de referencia ARMoWoT.

#### 4.1.10. Paso 10: Actualizar el repositorio con los nuevos Plugins creados

Por último, se almacenan los ocho *Plugins* generados en el los pasos anteriores, los cuales superaron las condiciones de reutilización y personalización requeridas. El almacenamiento se realiza en el siguiente repositorio:

<https://bitbucket.org/account/user/ariotteam/projects/ARMOWOT>

Y cumpliendo con la documentación de los *Plugins* se actualiza el tutorial establecido en la siguiente página web:

<https://armowot.000webhostapp.com/>

#### 4.1.11. Paso 11: Usar Plugin en el nuevo proyecto AR

Después de haber desarrollado todos los pasos del modelo, se procede a iniciar la construcción del proyecto AR para la WoT, del escenario de Domótica. Claro está utilizando



lo que actualmente se encuentra en el repositorio. A continuación, se describe el proceso de desarrollo de la interfaz:

## 4.2. CONSTRUCCIÓN DEL PROYECTO AR PARA LA WOT

### 4.2.1. Análisis

#### 4.2.1.1. Contextualización.

##### CONTEXTO.

El proyecto está enmarcado como subproyecto de la tesis de doctorado del PhD Miguel Ángel Niño Zambrano [11], y soportado a nivel de hardware sobre el escenario de interacción semántica para la IoT propuesto por Riobamba y Guerrero [17], y mejorado por Y.A.Pabón-Guerrero and L.J.Rojas-Bolaños [41].

El usuario objetivo para el cual está dirigido el presente proyecto son los desarrolladores, quienes en búsqueda de ahorro de tiempo y de soluciones adecuadas, desean generar óptimos modelos gráficos que representen los diferentes sensores, actuadores y servicios simulados o físicos, que hacen parte de un objeto inteligente. Sin embargo, a pesar de que no hacen parte del enfoque investigativo, se tiene un segundo usuario, el cual es un usuario final, quien usara la interfaz AR para observar los datos que arrojan los objetos inteligentes, visualizados mediante animaciones gráficas creadas por los desarrolladores mediante *Plugins* implementados bajo la guía del modelo de referencia propuesto en el capítulo 3.

El enfoque de solución para la presente investigación, es la creación de un modelo de referencia que contenga una guía para la creación de *Plugins* reutilizables y personalizables, que representen gráficamente mediante realidad aumentada los objetos inteligentes de la web de las cosas, otorgando como producto un desarrollo con lineamientos, pautas, recomendaciones y buenas prácticas de desarrollo.

##### ARQUITECTURA.

El estudio realizado por Niño-Zambrano [11], establece una arquitectura que hace uso de un índice semántico para contextualizar la información y los servicios que van a compartir los recursos IoT, sin embargo la interfaz no usa un acceso al índice semántico, debido a que no incluye botones con interacción entre objetos inteligentes, simplemente se abstrae la información de los diferentes objetos inteligentes y genera las representaciones gráficas de aquellos datos. Teniendo en cuenta la vista estática de la arquitectura de interacción semántica del autor (Ilustración contenida en el ANEXO A) donde se observa el desarrollo para la interfaz AR propuesto para el escenario de Domótica, se ubica principalmente en la capa de interacción, haciendo algo de uso de la capa de aplicación.

- **Capa de Interacción:** Para la presente tesis esta capa de consiste en la interacción de los IO, sus datos y las personas mediante sus dispositivos electrónicos, los cuales hacen uso de la técnica en AR para observan los datos (ver Ilustración 21).

### CAPA DE INTERACCIÓN

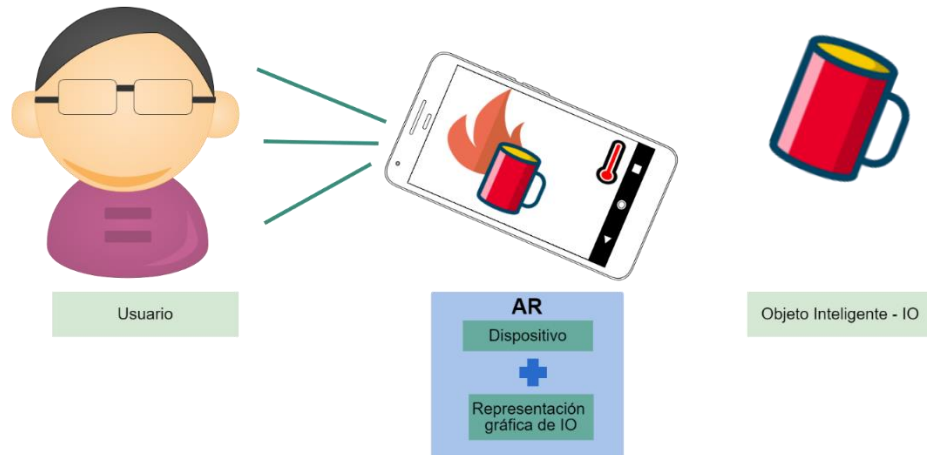


Ilustración 21 - Capa de interacción

- **La capa de aplicación:** Describe todo el procesamiento interno de los IO con protocolos de comunicación e inclusive el uso de un índice semántico para contextualizar la información de los distintos IO que hacen parte del escenario.

### ESCENARIO IoT.

La construcción del proyecto AR para la WoT, se realizara mediante el escenario de interacción semántica para la IoT propuesto por Riobamba y Guerrero [17], y mejorado por Y.A.Pabón-Guerrero and L.J.Rojas-Bolaños [41]. En general el escenario se compone de los siguientes recursos:

- 3 - Raspberry Pi3<sup>21</sup> equipadas con sistema operativo Linux<sup>22</sup>.
- 3 - Tarjetas Grove.
- 3 - Sensores Grove (sensor de luz, temperatura, humedad).
- 4 - Actuadores (led, ventilador, calefactor, riego simulado por un buzzer).

El escenario que se va representar gráficamente contiene tres recursos IoT, como se puede observar en la Ilustración 22, los cuales son OI definidos como:

- **Regulador de Luz:** Contiene un sensor de iluminación y una lámpara.
- **Regulador de Temperatura:** Contiene un sensor de temperatura, un actuador calefactor (emulado por un secador de aire caliente) y un actuador ventilador.

<sup>21</sup> <https://www.raspberrypi.org/>

<sup>22</sup> <https://www.linux.org/>

- **Regulador de Humedad:** Contiene un sensor de humedad y un dispositivo de riego (emulado por un buzzer).

Los anteriores objetos inteligentes contienen distintos sensores y actuadores, que a su vez tienen servicios los cuales se van a representar gráficamente aplicando la técnica de realidad aumentada.



*Ilustración 22 - Escenario de Domótica.*

Estos recursos se encuentran implementados sobre una tarjeta de desarrollo diseñada para la IoT, su nombre es Raspberry Pi 3, cuya características principales es contar con un microprocesador que soporta sistemas operativos orientados a la IoT, además de soportar comunicación vía Ethernet o Wifi entre otras características que en conjunto definen los IO respectivos. Para este caso en particular se utilizó como sistemas operativo Linux y el lenguaje de programación Python para desplegar las funcionalidades del escenario, además del Grove Starter kit de Seed Studio para desplegar los recursos físicos.

#### **4.2.1.2. REQUERIMIENTOS.**

##### **DEFINICIÓN DEL CLIENTE/DESARROLLADOR**

En este contexto y con el escenario IoT definido, se define el rol de cliente/final al PhD Miguel Ángel Niño Zambrano, cuyas funciones son generar los requerimientos de la aplicación, discutir todos los avances y generar la retroalimentación a lo largo del proceso de desarrollo para validarlo.

Como desarrolladores se tienen a los estudiantes Yuri Viviana Daza Gaviria y Juan Felipe Valencia Mosquera, del programa de ingeniería de sistemas de la Universidad del Cauca, cuyas funciones son llevar a cabo la captura de requerimientos, el desarrollo de



los prototipos funcionales para la creación de los modelos gráficos pertenecientes al escenario propuesto, en búsqueda del cumplimiento de los objetivos del presente trabajo.

## **CAPTURA DE REQUERIMIENTOS**

Para la captura de requerimientos se realizan historias de usuario. En este caso se resalta que el usuario cumple el rol de desarrollador, quien busca generar modelos gráficos a partir del modelo de referencia ARMoWoT presente en el anterior capítulo, representando el escenario<sup>23</sup> compuesto por tres objetos inteligentes (reguladores de luz, temperatura y humedad). Las representaciones gráficas a generar, se realizarán mediante modelos gráficos 2D o 3D, en un contexto<sup>24</sup> de aplicación de Domótica presente en una oficina.

Para la captura de requerimientos se seleccionan las historias de usuario, las cuales buscan identificar el flujo de información a través de la interacción del usuario con el escenario definido, para encontrar que las representaciones gráficas que se realizan sean adecuadas y logren en el usuario final una mejor abstracción de la información.

- **Historia de Usuario 1. Reutilización de Plugin para manipular modelos gráficos:** Yo como desarrollador de modelos gráficos en 2D o 3D, necesito uno o varios prefabricados reutilizables, que me permita manipular modelos gráficos mediante la técnica de realidad aumentada para representar la información de los objetos inteligentes propuestos en un escenario IoT.
- **Historia de Usuario 2. Personalización de Plugin para manipular modelos gráficos:** Yo como desarrollador de modelos gráficos en 2D o 3D, necesito uno o varios prefabricados personalizables, que me permita usarlos mediante la técnica de realidad aumentada y poder modificarlos según mis preferencias, para representar la información de los objetos inteligentes propuestos en un escenario IoT.
- **Historia de Usuario 3. Representaciones gráficas de los objetos inteligentes IoT:** Yo como usuario final necesito observar en mi dispositivo móvil una interfaz en 2D o 3D que represente la información de los sensores, actuadores y servicios, de los escenarios de interacción semántica IoT, mediante la técnica de realidad aumentada para tener acceso constante y claro de los datos arrojados.

## **ASIGNACIÓN DE TIEMPOS**

Esta investigación contiene actividades las cuales poseen un tiempo de desarrollo diferente a las que requiere la creación del proyecto AR para los IO, por otra parte, el desarrollo de la interfaz AR para los IO se realiza paralelamente a la comprobación de la efectividad del modelo de referencia ARMoWoT. Además, se presume que los usuarios

---

<sup>23</sup> Espacio físico o virtual contenido por diferentes dispositivos electrónicos conectados a la internet, más conocidos como objetos inteligentes, los cuales otorgan servicios a un usuario.

<sup>24</sup> El contexto se refiere al dominio de aplicación e intereses de los usuarios acerca de los servicios que puede proveer un objeto. Ejemplo: Utilizar un sensor de temperatura para medir la temperatura de una persona o utilizarlo para medir la temperatura de un cuarto en una casa, dónde se cambia tanto la entidad de interés: de persona a cuarto, como también el contexto: de salud a climático.





desarrolladores tienen las capacidades y los conocimientos necesarios, para que les tome poco tiempo apropiarse del escenario IoT y los recursos (*Plugins*) generados por ARMoWoT, construyendo así la interfaz AR para los IO.

## 4.2.2. Diseño

### 4.2.2.1. Establecimiento de tecnologías.

#### **TECNOLOGÍAS ESTABLECIDAS POR LA ARQUITECTURA Y EL ESCENARIO IOT**

Observando las características de los entornos de desarrollo descritas en la sección de investigación tecnológica del ANEXO B, se selecciona el entorno de desarrollo Unity3D debido a que, el entorno Unity3D es uno de los entornos más usados actualmente para la creación de proyectos relacionados con IoT y realidad aumentada, lo anterior se soporta con base en los trabajos relacionados presente en el marco teórico de la presente investigación, contando con los beneficios de que el entorno Unity3D cuenta con un total de 27 plataformas exportables entre las cuales están (iOS, Android, Windows Phone, Mac, Linux, Samsung Smart tv, Microsoft Hololens, Xbox One, entre otros), además de los dos beneficios más importantes de contar con conexión con la IoT, es decir, el entorno soporta los protocolos IP que se requieren para llevar a cabo la conexión con los objetos conectados a la internet y finalmente y no menos importante, el entorno posee un gran soporte para crear proyectos con realidad aumentada.

Es necesario realizar una convención acerca del nombre genérico de cada objeto para no dar espacio a confusiones durante la elaboración de los prototipos:

- **Objetos Inteligentes – OI:** Hacen referencia a los tres reguladores presentes en el escenario definido, contienen los recursos propios de cada regulador.

Por otra parte, los mensajes entre los IO y la representación virtual de los IO se realizan mediante los protocolos MQTT (Message Queue Telemetry Transport) y API REST.

Se puede concluir que, bajo un escenario funcional, su arquitectura base y demás tecnologías que implementa, se tomaría alrededor de un mes para la apropiación y acople.

#### **TECNOLOGÍAS A UTILIZAR PARA EL DESPLIEGUE DE LA INTERFAZ AR**

Una vez observadas las herramientas descritas en el ANEXO B, se selecciona la herramienta AR Vuforia debido a que, la herramienta es de tipo libre con opción de una SDK<sup>25</sup> comercial, aplicable o compatibles con entornos como iOS, Unity3D, Windows Mobile entre otros, adicionalmente posee un API de contenido siendo todas estas características muy importantes a la hora de crear una aplicación basada en realidad aumentada.

<sup>25</sup> SDK – software development kit, conjuntos de herramientas y librerías de código abierto (Open source) para soporte en el desarrollo de software que le permite al desarrollador la creación de aplicaciones para un sistema concreto





Por lo anterior, se crea la combinación del entorno y herramienta AR para crear los modelos gráficos de los objetos inteligentes del escenario propuesto en el presente estudio de caso.

#### 4.2.2.2. Plan de prototipos.

Según el estudio del escenario de domótica, se define realizar dos prototipos, donde el prototipo 1 resuelve las funcionalidades básicas del escenario definido implementando representaciones de los IO por medio de realidad aumentada, y el prototipo 2 refina la interfaz adicionándole una vista principal a las representaciones. Los servicios que brinda el escenario son:

- **Regulador de Temperatura:**
  - Mostrar Plugin 3D Actuador Calefacción.
  - Mostrar Plugin 3D Actuador Ventilador.
  - Mostrar Plugin 2D Temperatura.
  - Mostrar Plugin 3D Volcán Temperatura.
- **Regulador de Humedad:**
  - Mostrar Plugin 3D Actuador de Riego.
  - Mostrar Plugin 2D Humedad.
  - Mostrar Plugin 3D Humedad.
- **Regulador de Luz:**
  - Mostrar Plugin 3D Actuador Señal.
  - Mostrar Plugin 2D Luz.
  - Mostrar Plugin 3D Luz.

#### 4.2.3. Construcción

##### 4.2.3.1. Solución general

En la fase de construcción se van a utilizar los *Plugins* desarrollados mediante ARMoWoT, para generar la interfaz AR del escenario de Domótica.

#### INTERFAZ FÍSICA

Para el desarrollo de la solución se tienen como recurso dos portátiles con webcam, un teléfono móvil con sistema operativo Android en modo desarrollador y el entorno de desarrollo Unity3D. De esta manera la visualización de los datos es de forma directa, donde el usuario final deberá sostener su dispositivo electrónico, apuntando hacia el identificador del objeto inteligente y que en la pantalla del dispositivo, aparezca todas las representaciones en 2D o 3D de la información de los IO.

#### COMUNICACIÓN CON EL ESCENARIO



El escenario tiene como sistema de comunicación entre los IO y la representación virtual de los IO, mensajes en MQTT descrito en la Ilustración 13 y API REST. Para conseguir la comunicación por MQTT se hace necesario incluir la librería de M2MQTT4Unity y para usar el protocolo de comunicación de API REST se incluye la librería de Unity llamada UnityEngine.Networking.UnityWebRequest<sup>26</sup>. Por ultimo para descifrar los datos de los sensores, los cuales llegan como texto plano en formato de XML, se utiliza la librería llamada System.Xml.Linq<sup>27</sup>.

#### 4.2.4. Prototipo 1

La interfaz implementada consta de 10 representaciones entre 2D y 3D. Las cuales se encuentran distribuidas entre los sensores y actuadores del escenario de Domótica de la siguiente manera:

- Sensor de Temperatura: Con una representación en 2D y 3D.
- Actuador Calefacción: Con una representación en 3D.
- Actuador Ventilador: Con una representación en 3D.
- Sensor de Humedad: Con una representación en 2D y 3D.
- Actuador Riego: El cual esta simulado por un buzzer, tiene una representación en 3D.
- Sensor de Luz: Con una representación en 2D y 3D.
- Acatador Bombilla: Con una representación en 3D.

La última representación es un icono, con forma de la cabeza de una flecha, la cual permite ocultar las representaciones en 2D, las cuales se ubican mediante un Canvas a la izquierda de la pantalla del dispositivo.

<sup>26</sup> <https://docs.unity3d.com/560/Documentation/Manual/UnityWebRequest.html>

<sup>27</sup> <https://msdn.microsoft.com/en-us/library/system.xml.linq%28v=vs.110%29.aspx?f=255&MSPPEror=-2147217396>

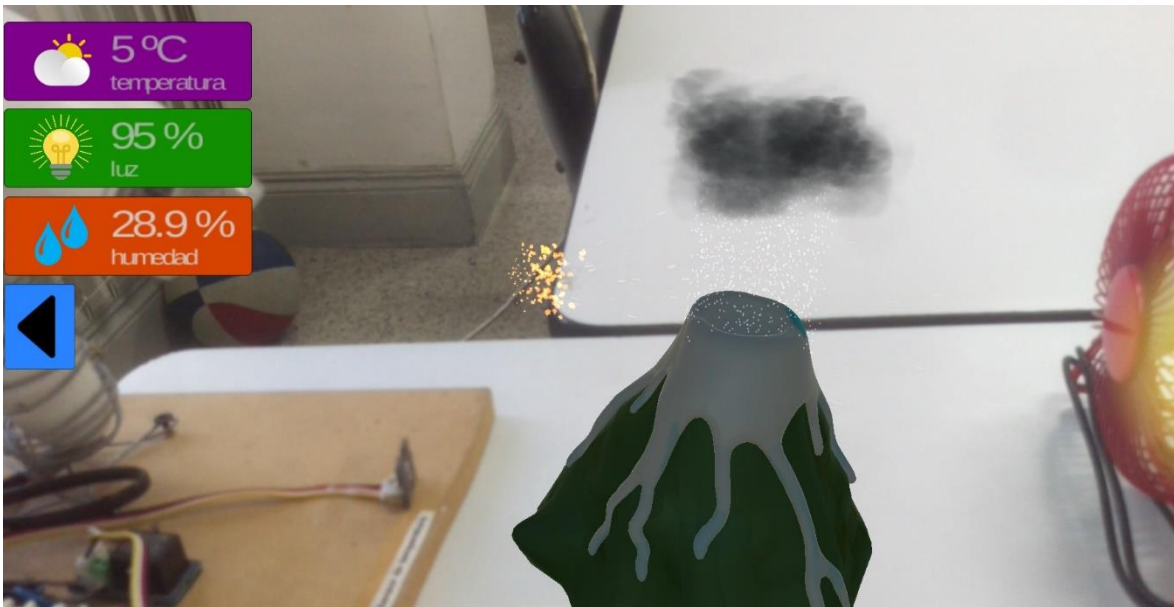


Ilustración 23 - Prototipo 1

#### 4.2.4.1. SOLUCIÓN PARA LA HISTORIA DE USUARIO 1

Inicialmente en la historia de usuario uno, se describe que el desarrollador de modelos gráficos en 2D o 3D, necesita uno o varios prefabricados reutilizables, que le permita usarlos mediante la técnica de realidad aumentada, para representar la información de los objetos inteligentes propuestos en un escenario IoT.

Lo primero que se realiza es crear el nuevo proyecto en Unity, prepara el entorno con el SDK de Vuforia<sup>28</sup> (Ver preparación en el ANEXO E). Paso siguiente se establece un requerimiento como prioritario, en donde se requiere importar una lista de *Plugins*, que me permita reutilizarlos en la nueva interfaz para el escenario de Domótica. Para ello ingresamos al siguiente repositorio:

<https://bitbucket.org/account/user/ariotteam/projects/ARMOWOT>

El cual contiene varias opciones de representación en AR para IO, creadas mediante el modelo de referencia ARMoWoT. Debido a lo anterior se procede a seleccionar todos los *Plugins* necesarios para la construcción del Proyecto AR. Por lo que listamos los siguientes *Plugins* que serán los usados en el escenario de Domótica, de acuerdo a cada IO:

- **Regulador de Temperatura:**
  - Plugin 3D Actuador Calefacción.
  - Plugin 3D Actuador Ventilador.
  - Plugin 2D Temperatura.
  - Plugin 3D Volcán Temperatura.

<sup>28</sup> La preparación del entorno puede variar dependiendo de la versión de Unity, esto debido a que la última versión ya integra la técnica de AR como una función más del motor de desarrollo. En caso contrario de no tener la última versión a la documentación de Vuforia: <https://developer.vuforia.com/downloads/sdk>

- **Regulador de Humedad:**
  - Plugin 3D Actuador de Riego.
  - Plugin 2D Humedad.
  - Plugin 3D Humedad.
- **Regulador de Luz:**
  - Plugin 3D Actuador Señal.
  - Plugin 2D Luz.
  - Plugin 3D Luz.

### Escenario 1:

Después se importan los recursos en el proyecto de Unity, donde se visualizara la siguiente arquitectura de carpetas:

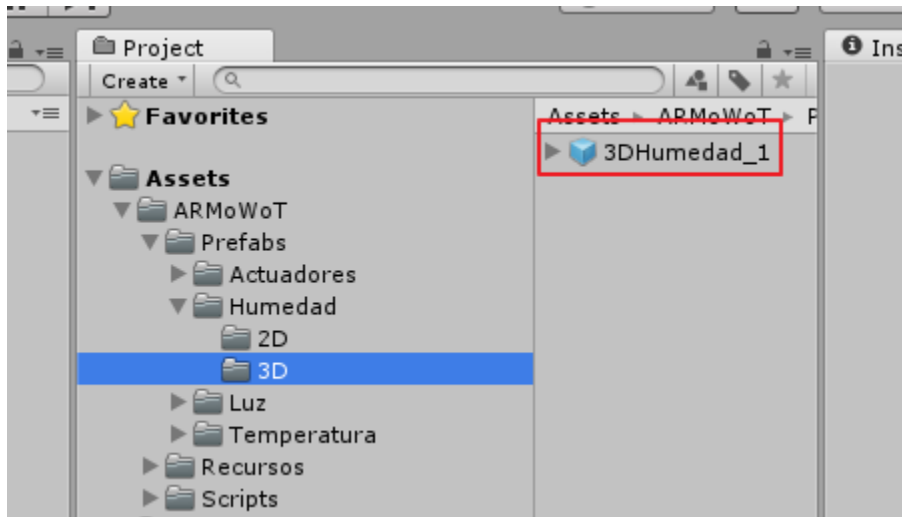
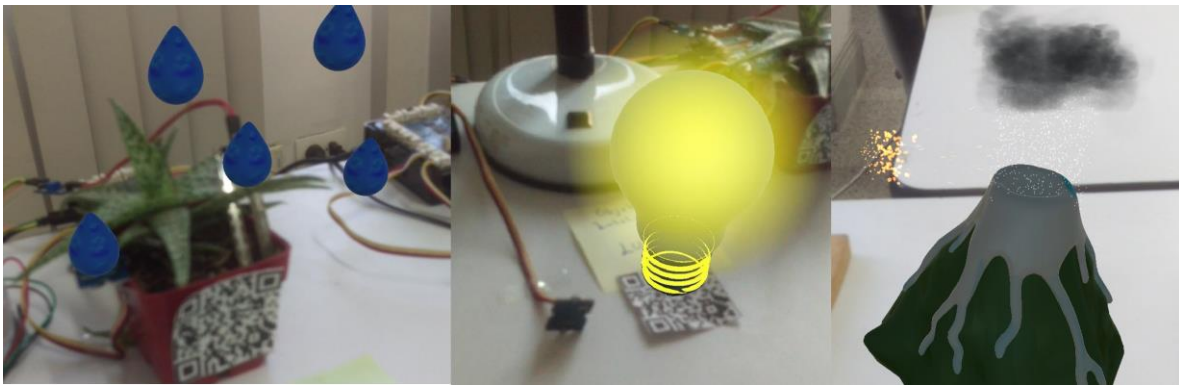


Ilustración 24 – Jerarquía de carpetas para los Plugins en el entorno de Unity 3D

En la imagen anterior (ver Ilustración 24) se puede observar que todos los *Plugins* se encuentran organizados dentro de la carpeta “ARMoWoT” y a su vez esta contiene tres carpetas: Prefabs (contiene las representaciones listas para usar en 2D y 3D), Recursos (Contiene texturas, animaciones, modelos 3D y todos los recursos gráficos necesarios para las representaciones) y Scripts (toda la lógica en base a comunicación y funcionalidad de las representaciones, tiene la misma organización que la carpeta de Prefabs). Por lo tanto se procede a tomar todos los elementos necesarios y colocarlos en el escenario del proyecto, con lo que se tendrá el siguiente resultado:



*Ilustración 25 - Representación 2D para sensores del escenario de Domótica.*



*Ilustración 26 - Representación 3D para sensores del escenario de Domótica.*

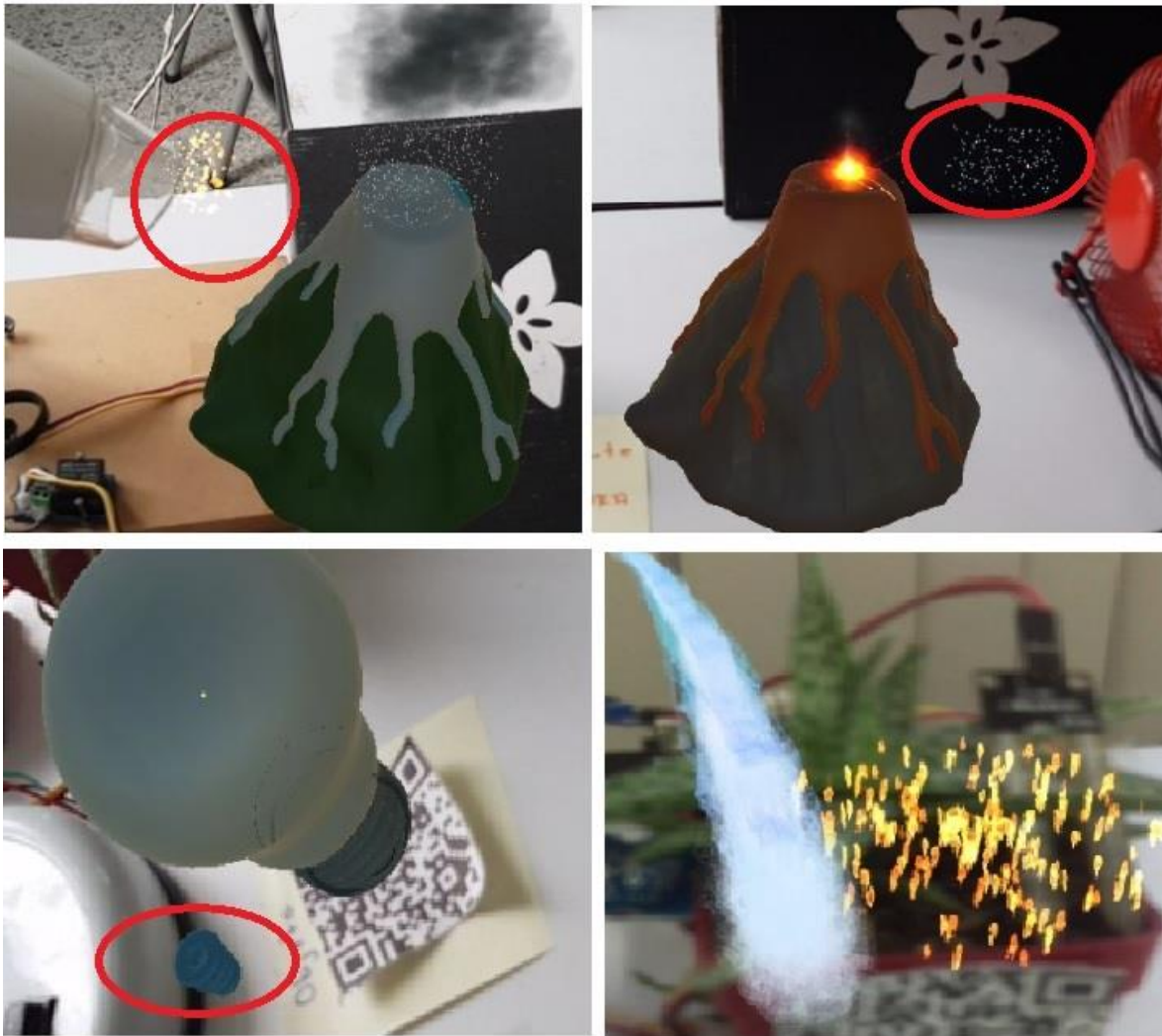


Ilustración 27 - Representación 3D para actuadores del escenario de Domótica (**Representación opcional de actuadores**).

#### 4.2.4.2. Solución para la historia de usuario 2

La HU2 describe que el desarrollador de modelos gráficos en 2D o 3D, necesita uno o varios prefabricados personalizables, que le permita usarlos mediante la técnica de realidad aumentada y poder modificarlos según las preferencias del desarrollador, para representar la información de los objetos inteligentes propuestos en un escenario IoT.

##### Escenario 1:

Teniendo en cuenta que en la anterior historia de usuario ya reutilizo todos los *Plugins* necesarios para las representaciones, se procede a personalizar sus colores, textura y tamaños, para acoplarlos a lo que se desea observar en el escenario. Para cumplir la consigna de personalización se presenta a continuación el enlace a la documentación que le permite al desarrollador tener una base para la personalización de los *Plugins*:



<https://armowot.000webhostapp.com/documentacion/>

## 4.2.5. Prototipo 2

La interfaz implementada se compone del prototipo 1, pero con algunas variaciones al **Escenario 1** desarrollado anteriormente. Estas variaciones son enfocadas en proporcionar algunos componentes 2D para establecer un escenario con una pantalla amigable y sencilla de manejar. Después se da paso a crear un **Escenario 2**, donde se establecer una navegación entre escenas y una presentación introductoria de la aplicación.

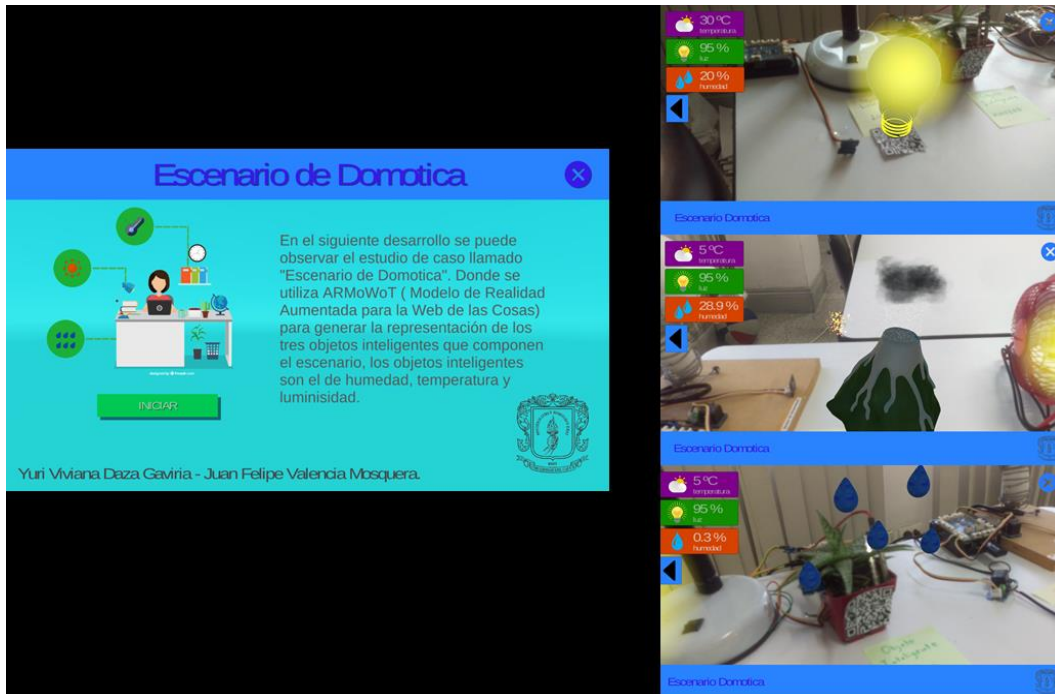


Ilustración 28 - Prototipo 2

### 4.2.5.1. Solución para la historia de usuario 3

En la HU3 el usuario final, observar en el dispositivo móvil personal una interfaz en 2D o 3D que represente la información de los sensores, actuadores y servicios, de los escenarios de interacción semántica IoT, mediante la técnica de realidad aumentada para tener acceso constante y claro de los datos arrojados.

Por tal motivo, se encuentra como requerimiento prioritario: Obtener y permitir la visualización de los datos que contienen los diferentes IO del escenario.

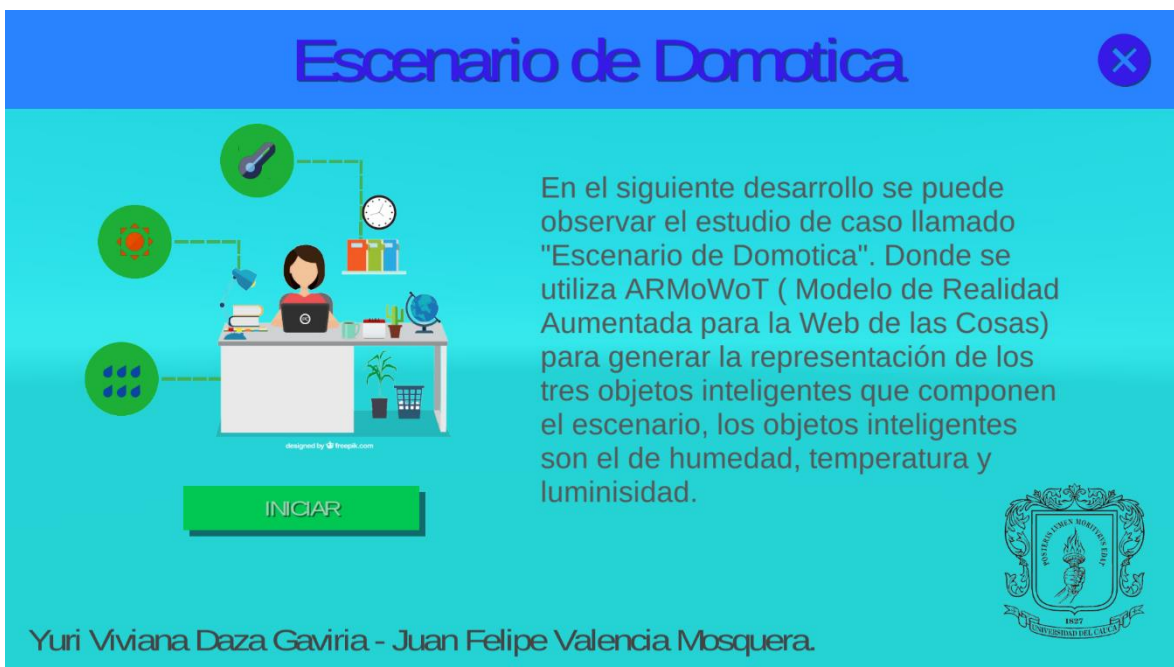


### **Escenario 1:**

Se procede a generar una escena principal en el que se agrega un botón posicionado en la derecha superior para ingresar en la escena 2. También se agrega un banner<sup>29</sup> en la parte inferior de la pantalla con el nombre del escenario y un icono de la Universidad del Cauca, para generar una estética más apropiada.

### **Escenario 2:**

Para la segunda pantalla, se crea un espacio de presentación para la interfaz en AR. La cual contiene una breve introducción e iconos alusivos al escenario y un botón de navegación, la cual inicia la captura de identificadores en AR.



*Ilustración 29 - Escenario 2, pantalla principal*

La fase de validación de los Plugin se realiza a continuación en el capítulo 5 de este documento.

<sup>29</sup> Espacio publicitario insertado normalmente en páginas web



## CAPÍTULO 5 EVALUACIÓN Y ANÁLISIS

### 5.1. FASE DE DEFINICIÓN DE LOS INDICADORES.

En la presente fase de definición de los indicadores, se realizará una caracterización de todos aquellos individuos con los cuales se llevará a cabo la prueba de los *Plugins* construidos en la presente investigación. Adicionalmente se detallará un plan de pruebas y los indicadores en búsqueda de responder a la pregunta de investigación planteada, la cual dice: **¿Cómo crear representaciones graficas de los objetos inteligentes en una interfaz AR para la WoT, que sean reutilizables y personalizables, de acuerdo a su contexto de aplicación?**, lo anterior se tendrá en cuenta con el objetivo de definir todo el proceso de evaluación.

#### 5.1.1. Caracterización del personal a evaluar

La caracterización facilita identificar la población donde se va a ejecutar las pruebas para comprobar los *Plugins* desarrollados con base en el modelo de referencia ARMoWoT. En el presente documento se toma en cuenta una muestra probabilística. Como primera etapa se escoge una población entre artistas y desarrolladores. Como segunda etapa se buscan características y/o conocimientos de la muestra seleccionada de artistas y desarrolladores. Teniendo en cuenta que debido a las herramientas software usadas para el desarrollo en base a ARMoWoT, se hace necesario establecer conocimientos básicos pero específicos a los participantes.

Por lo anterior, se define en el ANEXO F, por medio de la **“Prueba Inicial (Pre-Test)”**, aquellas características que deben poseer los individuos ideales para llevar a cabo la comprobación total del modelo de referencia.

Con la anterior caracterización se desean obtener personas con conocimientos básicos en el manejo de Unity 3D, gráficos en 2D, modelos gráficos en 3D y conocimiento teórico acerca de los sensores y actuadores. Por otro lado, el manejo del SDK de Vuforia es algo alternativo, esto debido a que en la actualidad la herramienta se encuentra incorporada a Unity y por lo tanto la curva de aprendizaje es muy baja, debido a eso se proporciona una capacitación acerca del uso de Vuforia en Unity a todos los participantes.

La descripción de la definición de los indicadores y respectivas encuestas a realizar se alberga en el ANEXO F, donde se tiene los parámetros condicionales para su selección. Como método de recolección de datos e indicadores a tener en cuenta a lo largo de esta evaluación se tiene:

- Se evalúa por medio de observación o entrevista los siguientes indicadores:
  - Reutilización del Plugin.
  - Personalización del Plugin.
- Se evalúa por medio de encuesta los siguientes indicadores:
  - Rapidez al momento de aplicar el Plugin.



- Facilidad de uso del Plugin.

## 5.2. FASE DE IMPLEMENTACIÓN DE INDICADORES.

En esta fase se realizará la planeación y los elementos necesarios para implementar la evaluación de los indicadores a los participantes. Los métodos de recopilación de datos y todos los pasos con los cuales se realizara la prueba se contienen en el ANEXO F. Teniendo en cuenta que el siguiente diagrama (ver Ilustración 30) muestra un recuento de las actividades a realizar a lo largo de la evaluación:

### 5.2.1. Diagrama de Gantt para el Plan de pruebas

Actividad		Tiempo (Minutos)	ACTIVIDAD									Encargado
			#1	#2	#3	#4	#5	#6	#7	#8	#9	
Descripción de la investigación realizada		23										
1	Indicar objetivos principales de la investigación.	5										Yuri
2	Indicar agenda y objetivos principales de la prueba.	5										Yuri
3	Indicar el modelo de referencia ARMoWoT.	10										Juan
4	Realizar la Prueba Inicial (Pre-Test).	3										Participantes
Capacitación para los participantes		10										
5	Capacitación AR con herramienta VuforiaSDK	10										Yuri
Realización de la Prueba		45										
6	Entregar documento en formato PDF y otros recursos	5										Yuri-Juan
7	Proceso de desarrollo de la prueba	40										Participantes
Realización de la Prueba		10										
8	Prueba Final para el participante	5										Participantes
9	Prueba de comprobación - Prueba Final (Post-Test)	5										Yuri-Juan
Duración total de la actividad de prueba		88										

Ilustración 30 - Diagrama de Gantt para el Plan de Pruebas

## 5.3. FASE DE ANÁLISIS Y RETROALIMENTACIÓN AL MODELO.

En esta fase se presentan y analizan los resultados obtenidos mediante la evaluación anterior, hecha a los participantes. Al final se decide si debe hacerse alguna modificación al modelo de referencia y/o a los *Plugins* con respecto a la experiencia de los participantes.

### 5.3.1. Presentación de los resultados

Como primera instancia se tiene en cuenta que la presentación de la prueba inicial, se hace necesaria para caracterizar a los participantes y comprobar que tenían los conocimientos esenciales para realizar la evaluación. Por lo tanto se anexan los resultados, como válidos para cada participante y se procede a analizar las pruebas que evalúan los indicadores centrales.

**EVALUACIÓN PARA:** Reutilización y personalización.

Para comprobar los indicadores de reutilización y personalización (con base en tamaño, color o textura) de los *Plugins*. Se tiene en cuenta la participación de 8 individuos, que de acuerdo a su caracterización tienen conocimientos y habilidades similares. Por lo tanto se evalúa el desarrollo realizado por los participantes por medio de la encuesta "**Prueba de comprobación**" de la cual se denotan los siguientes resultados.

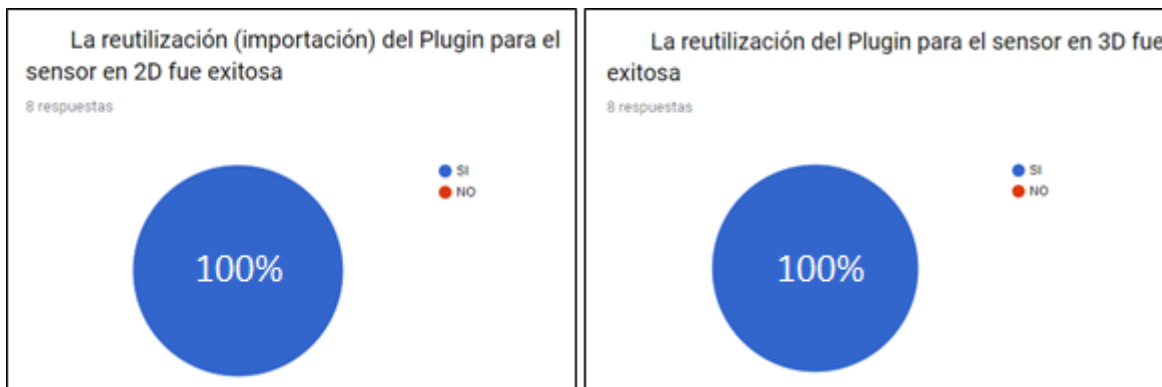


Como objetos inteligentes a ser probados se establecieron cuatro, en los que se encuentran: IO Temperatura, IO Nivel, IO Humedad, IO Luz. Por lo que se distribuyen igual cantidad de objetos inteligentes y *Plugins* a cada dos personas, de la siguiente manera:

SUJETO DE PRUEBA	IO A IMPLEMENTAR	CONTEXTO	PLUGIN 2D	PLUGIN 3D
Sujeto 1	IO Temperatura	Domótica	Plugin 2D para Temperatura	Plugin 3D para Temperatura
Sujeto 2		Contaminación ambiental		
Sujeto 3	IO Nivel	Domótica	Plugin 2D para Nivel	Plugin 3D para Nivel
Sujeto 4		Contaminación ambiental		
Sujeto 5	IO Humedad	Domótica	Plugin 2D para Humedad	Plugin 3D para Humedad
Sujeto 6		Contaminación ambiental		
Sujeto 7	IO Luz	Domótica	Plugin 2D para Luz	Plugin 3D para Luz
Sujeto 8		Contaminación ambiental		

Tabla 7 - Distribución de Pruebas para los participantes

El objetivo de la anterior distribución es permitir que cada sujeto pueda experimentar la reutilización en proyectos diferentes y con diferentes contextos. Por lo que podemos ver los siguientes indicadores:

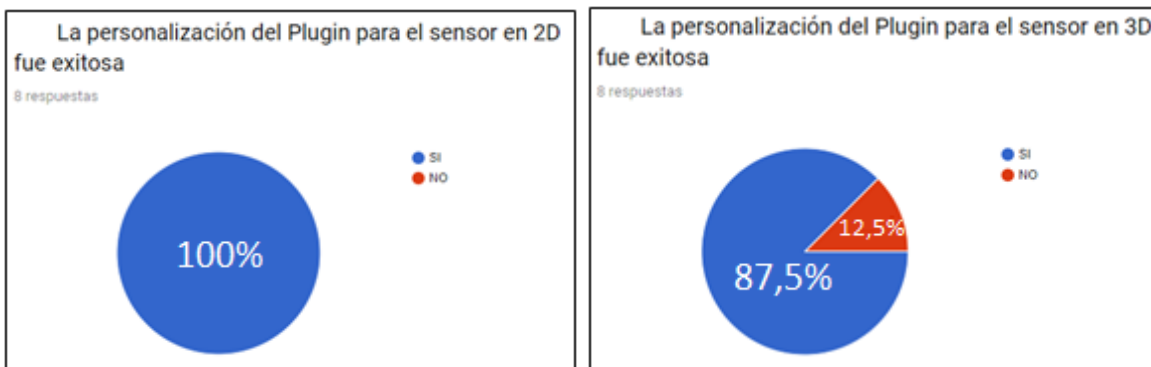


Gráfica 1 - Porcentaje de Reutilización de Plugins

El estudio de la capacidad de personalización y reutilización que obtuvieron los plugins, fue evaluado mediante el método de observación. Esto es debido a que la personalización y reutilización podrían considerarse conceptos algo subjetivos, de los cuales cada persona podría llegar a tener una forma distinta de entenderlos y considerarlos. Por lo tanto, si se permitiera que los sujetos encuestados determinaran ¿Que tan reutilizables y personalizables son los plugins creados?, posiblemente generaría ruido en cuanto a los resultados. De esta manera no se podría determinar si realmente los plugins cumplieron con el objetivo de caracterización buscado inicialmente.

Durante la prueba, se observó que cuando los sujetos realizaban la importación del plugin que necesitaban desde el repositorio al proyecto de su escenario, los ocho sujetos lograron reutilizar correctamente los plugins necesarios para los IO de temperatura, nivel, humedad y luz, sin generar algún inconveniente.

Por lo anterior, se determina que el indicador de reutilización, con respecto al entorno de desarrollo de Unity3D fue exitosa. Ahora pasamos a ver los resultados para la personalización, en donde se distribuye igual cantidad de *Plugins* para cada pareja, como se puede ver en la Tabla 7, para que modifiquen las características de tamaño, color o textura en los *Plugins* según preferencias personales.

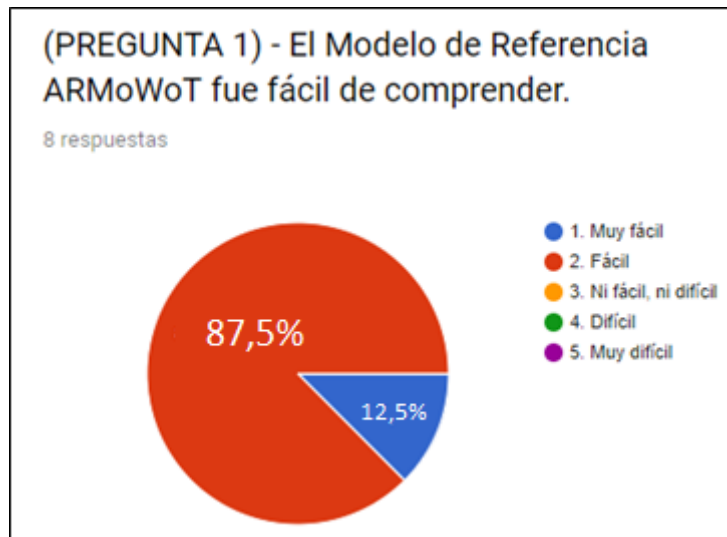


Gráfica 2 - Porcentaje de Personalización de Plugins

Como se puede observar, la personalización para los *Plugins* de tipo 2D se realizó satisfactoriamente y para la personalización en 3D, se presentó un 12.5% equivalente a una persona, la cual no pudo realizar la actividad de personalización. Realizando un análisis de lo sucedido, se debe entender que si bien los *Plugins* son personalizables, son susceptibles a cambios en sus componentes internos, por lo tanto si no se sigue bien el instructivo de uso, se puede incurrir en daños al Plugin haciendo que la personalización quede desacoplada de lo que se espera ver visualmente.

**EVALUACIÓN PARA:** Rapidez y facilidad de uso.

Debido a que no se está evaluando una interfaz ya terminada sino, la aplicación de representaciones ya prediseñadas, para ser reutilizada y personalizadas al gusto de los usuarios desarrolladores. Se genera un formulario de preguntas establecida por medio de la escala Likert, para conocer aspectos de satisfacción o insatisfacción frente a los *Plugins* desarrollados con ARMoWoT. La encuesta es realizada a los 8 participantes, una vez han concluido su prueba técnica en el entorno de Unity 3D. Como se puede observar en la Gráfica 3 el nivel de comprensión con respecto al modelo ARMoWoT se establece entre los niveles de fácil a muy fácil, con una tendencia del 87.5% a ser fácil de entender.



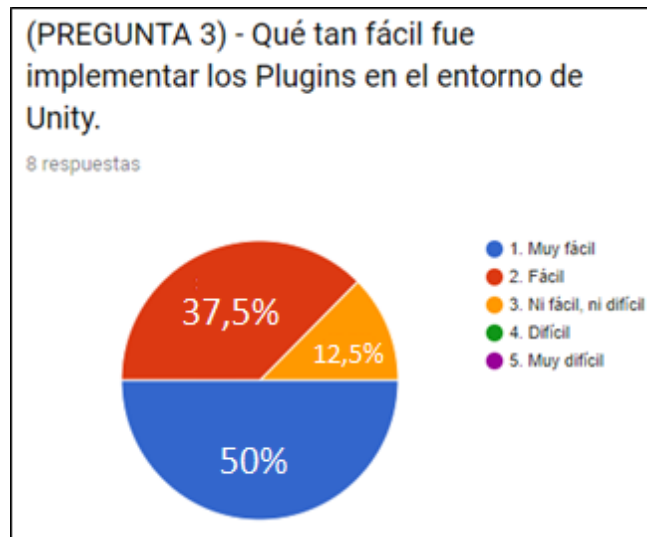
Gráfica 3 - Nivel de comprensión sobre ARMoWoT.

Ahora bien para la pregunta dos (ver Gráfica 4), la cual se enfoca a la facilidad de los tutoriales para ser entendidos, se puede observar una tendencia que esta entre “Muy fácil” hasta “Ni fácil, ni difícil”, dándonos a ver una tendencia a fácil con el 62.5%. Aunque la métrica indica favorabilidad siempre hay formas de mejorar los tutoriales que se fabrican para comprender el nivel de personalización que se le puede otorgar a cada Plugin. Como puede ser videos o presentaciones interactivas.



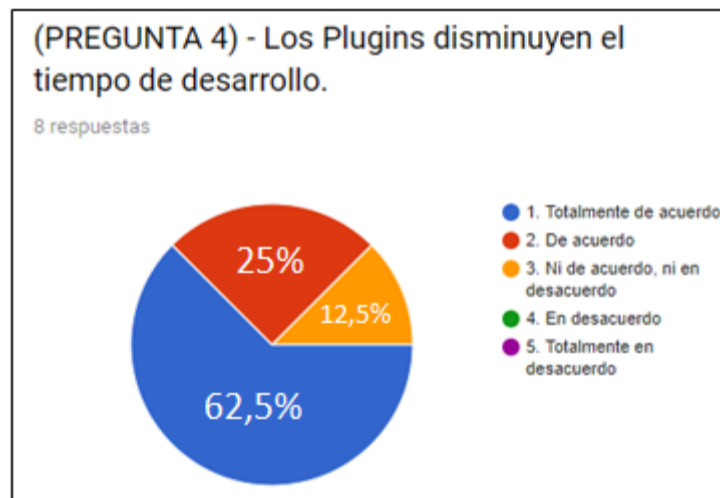
Gráfica 4 - Tutoriales para Plugins fáciles de entender.

El enfoque para esta pregunta es comprender que tan difícil o fácil fue para el usuario comprender los *Plugins* y aplicarlos en el entorno de desarrollo. Se obtiene una tendencia del 50% (ver Gráfica 5) donde se da a entender que fue Muy fácil implementarlo, esto es gracias al entorno de desarrollo, el cual proporciona toda la infraestructura para que la implementación no sea tan robusta y facilita la curva de aprendizaje de los *Plugins*.



Gráfica 5 - Facilidad al implementar Plugins en Unity 3D.

Hasta ahora solo se había tenido en cuenta el indicador sobre facilidad de uso, ahora se procede a evaluar el indicador de rapidez, que puede ser variable de acuerdo a la percepción de cada individuo, por lo tanto se realiza una afirmación para generalizar el concepto de rapidez de cada sujeto de prueba y obtener una respuesta más transparente. Como se puede observar en la Gráfica 6 la mayoría con un porcentaje del 62.5% coincide en que si existe un ahorro de tiempo al utilizar modelos encapsulados por medio de un Plugin.



Gráfica 6 - Nivel de ahorro de tiempo usando Plugins.

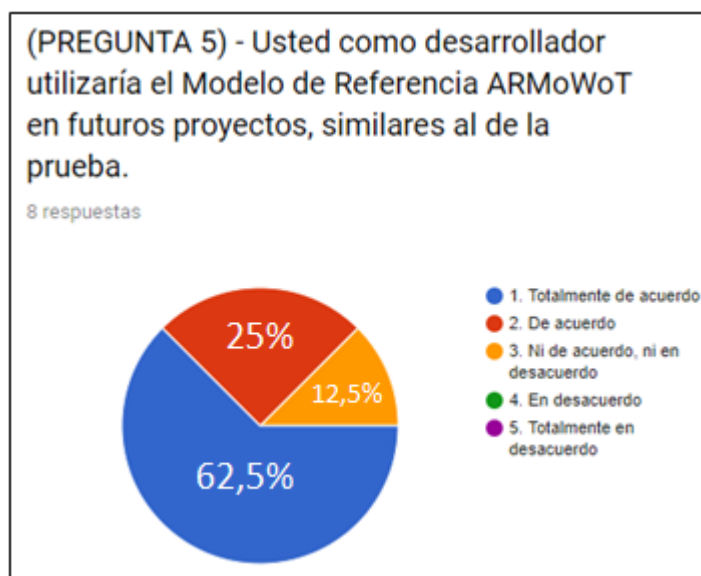
Por último se realiza una pregunta enfocada a la utilidad real del modelo (ver Gráfica 7), en otros proyectos que sean realizados por nuestros sujetos de prueba, el 62.5% de los individuos encuestados tienen un firme interés en aplicar el modelo, pero existe un 12.5% que aún no está convencido de usarlo, y se determinó que ese 12,5% no estaban convencidos, debido a que consideraban muchos pasos para un desarrollo ágil, sin embargo se aclara que el modelo ARMoWoT es un conjunto de lineamientos que bien el





desarrollador tiene la libre elección de seleccionar los pasos que crea convenientes seguir en búsqueda de generar una buena representación grafica de los objetos inteligentes de su escenario.

Debido a todo lo anterior, se deduce que el modelo debe seguir evolucionando de acuerdo a las necesidades de los desarrolladores.



Gráfica 7 - Nivel de aplicación del modelo en otros proyectos

Por último se genera una tabla de resultados, de acuerdo a los parámetros dictados por la escala de Likert y teniendo en cuenta los puntos proporcionados a cada respuesta. Siendo las respuestas más favorables representadas por un 5 y las menos favorables representadas por un 1. A continuación se proceden a realizar los cálculos de la encuesta total:

Sujeto de prueba	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Promedio
Sujeto 1	4	4	3	5	5	4.2
Sujeto 2	4	5	4	3	5	4.2
Sujeto 3	4	3	5	4	3	3.8
Sujeto 4	4	3	5	5	4	4.2
Sujeto 5	5	4	4	5	5	4.6
Sujeto 6	4	4	5	5	5	4.6
Sujeto 7	4	4	4	5	5	4.4
Sujeto 8	4	4	5	4	4	4.2
<b>PROMEDIO TOTAL</b>						<b>4.275 = 4.3</b>

Tabla 8 - Consolidado de la Evaluación.

De acuerdo a la anterior tabla se puede proceder a establecer el puntaje dentro de la escala de Likert.



Ilustración 31 - Resultados de la Escala de Likert.

Por lo tanto se puede ver que de acuerdo a la muestra tomada para la realización de la prueba, fue satisfactoria su implementación. Cumpliendo con los indicadores de reutilización, personalización, facilidad de uso y rapidez al implementar *Plugins*, teniendo en cuenta que se realizó con el entorno de desarrollo Unity y el SDK de Vuforia.

### 5.3.2. Retroalimentación al modelo ARMoWoT

Durante el desarrollo de la presente investigación, y bajo los resultados obtenidos en la realización del estudio de caso, se determinaron muchas necesidades entre las cuales se destacan:

- La necesidad de establecer buenas prácticas a la hora de generar tutoriales, documentación o guías que permitan conocer el nivel de personalización que pueden obtener los *Plugins* desarrollados bajo el modelo de referencia ARMoWoT.
- La necesidad de establecer una jerarquía de carpetas a la hora de codificar las soluciones de los *Plugins*.
- La necesidad de contar con una vista de edición exclusiva para personalizar los *Plugins* que se desarrollan en los diferentes entornos y herramientas.
- Y la necesidad de crear espacios de comunidad abierta, como una página web donde se depositen tutoriales e información esencial para aplicar los *Plugins* en diferentes contextos.

Todo lo anterior se considera importante, con el objetivo de tener buenas representaciones gráficas desarrolladas mediante *Plugins*, sin embargo, es de resaltar que debería establecerse algún tipo de caracterización de las personas que podrían utilizar el modelo ARMoWoT creado. Debido a que posiblemente se requiera de un perfil como desarrollador, para que sea mucho más sencillo el seguimiento de los pasos definidos en el modelo.



## CAPÍTULO 6 CONCLUSIONES Y TRABAJO FUTURO

### 6.1. CONCLUSIONES

#### 6.1.1. CUMPLIMIENTO DE OBJETIVOS

##### OBJETIVO GENERAL:

Proponer un modelo de referencia para la creación de modelos gráficos de objetos inteligentes en realidad aumentada para la WoT, a través del desarrollo de *Plugins* personalizables y reutilizables en diferentes contextos de la WoT.

##### OBJETIVOS ESPECÍFICOS:

- Definir un conjunto de lineamientos para la creación de modelos gráficos de objetos inteligentes, de tal manera que permita:
  - i. Establecer la representación gráfica del IO de acuerdo a las características tales como: dispositivos, sensores, actuadores y servicios.
  - ii. Definir un mecanismo para su personalización y reutilización en interfaces AR, de acuerdo a su contexto de aplicación.
  - iii. Proponer la forma de crear *Plugins* para AR, que recojan las representaciones gráficas definidas del IO.
- Desarrollar una interfaz de AR para la WoT en estudios de caso que serán seleccionados en el transcurso del proyecto.
- Evaluar la capacidad de personalización y reutilización de los *Plugins* en otros contextos de aplicación.

##### CUMPLIMIENTO:

- Con el producto obtenido en la investigación del modelo de referencia ARMoWoT y los lineamientos descritos en cada paso, se considera cumplido el objetivo específico número uno, donde se propone la definición de un conjunto de lineamientos para la creación de modelos gráficos de objetos inteligentes de tal manera que permitan establecer la representación gráfica de objetos inteligentes de acuerdo a las características de sus componentes como sensores, actuadores y servicios. Además, se realizó la definición de un mecanismo de personalización y reutilización en interfaces de realidad aumentada, implícitos en el paso a paso que permiten crear y/o reutilizar *Plugins* que recojen representaciones gráficas de los distintos objetos inteligentes de la web de las cosas.
- Posteriormente a la creación del modelo de referencia, se realizó una prueba preliminar del modelo, obteniendo una primer interfaz en AR, que mediante *plugins* personalizables y reutilizables representaban gráficamente en 2D y 3D la temperatura y el nivel del líquido contenido en un termo. De igual manera, en el estudio de caso, se desarrollo una segunda interfaz AR la cual otorgó continua



retroalimentación al modelo, rubusteciendolo y generando a la vez mejores practicas de desarrollo cuando se trata de construir interfaces graficas que representen datos de los distintos objetos inteligentes.

- Finalmente, se realizo la evaluación de las características de personalización y reutilización que deben tener los *Plugins* fabricados bajo el modelo. Todo llevado a cabo por medio de los métodos de observación y métodos de encuestas, arrojando resultados positivos y permitiendo concluir que el modelo de referencia tiene grandes ventajas cuando se trata de crear modelos graficos que representen información de los sensores y actuadores pertenecientes a los objetos inteligentes.

### 6.1.2. CONCLUSIONES EN CUANTO A PRODUCTOS OBTENIDOS

- Se ha propuesto un modelo de referencia genérico para las representaciones graficas en 2D o 3D de los objetos inteligentes en realidad aumentada para la WoT, enfocados en la reutilización y la personalización, con el fin de que puedan ser utilizados en diferentes contextos.
- Se construyeron 13 *Plugins* utilizando el modelo de referencia propuesto, con cualidades para ser reutilizados en el entorno de Desarrollo Unity 3D y ser personalizables, por medio de una ventana de edición, en características como el tamaño, color o textura.
- Se construyeron dos interfaces de realidad aumentada: La primera para el Termo Inteligente, el cual uso un total de 5 *Plugins* junto con los programas para la navegación por medio de escenas. La segunda interfaz, en donde su desarrollo fue más extenso, se utilizó un total de 10 *Plugins*, reutilizando a su vez 2 *Plugins* usados en el desarrollo del Termo Inteligente.
- Se usó un repositorio de acceso público donde se almacenan los *Plugins* y una página web, la cual documenta el uso de los *Plugins* por medio de tutoriales textuales y en video. También contiene información acerca del modelo de referencia ARMoWoT y como se usó para construir los *Plugins*.
- Se desarrolló un programa (clase) la cual utiliza librerías de terceros con el fin de facilitar la comunicación entre los objetos de la WoT, generando a su vez otra característica de personalización, ya que la lógica permite escoger entre dos protocolos MQTT y API REST.

### 6.1.3. CONCLUSIONES EN CUANTO A LA INVESTIGACIÓN

- En la evaluación realizada sobre la reutilización y personalización de los *Plugins* construidos con el modelo ARMoWoT propuesto, cada usuario fue capaz de reutilizar y personalizar a gusto propio datos como tamaño, color o textura, obteniendo satisfacción en los desarrolladores que realizaron la prueba. Sin embargo, algunos de los evaluadores no realizaron todas las actividades, debido a



que los *Plugins* son vulnerables a cambios por parte del usuario. Otorgando de esta manera retroalimentación referente a que tan robusta debe ser la programación para evitar fallos y convertir los *Plugins* en algo más automático.

- A partir de la experimentación, bajo los indicadores de facilidad de aplicación y uso de los *Plugins* fabricados, se obtuvo un puntaje de 4.3 sobre 5 en la escala Likert. Por lo tanto, se concluye que el modelo obtuvo una tendencia positiva, al ser capaz de orientar a los desarrolladores bajo lineamientos de construcción bien definidos para representar gráficamente en 2D y 3D los datos de sus dispositivos. De igual manera, se pudo evidenciar que *la percepción de ahorro de tiempo (rapidez)* en el desarrollo fue considerable, con respecto al que tocaría invertir si se realizara el desarrollo desde cero. Por lo tanto el modelo ARMoWoT se considera un buen producto final que bajo lineamientos, buenas practicas de desarrollo y mecanismos de personalización y reutilización genera un buen aporte al sector software que se dedica a generar representaciones gráficas de información de toda la IoT centralizado en la web WoT, sin embargo el modelo de referencia ARMoWoT, tiene algunas falencias y se denotan en cuanto a que una desarrollador posiblemente lo considere algo robusto por la cantidad de pasos a seguir y posiblemente no tenga en cuenta pasos importantes que garanticen un buen producto final.
- La experiencia al utilizar el modelo propuesto, varía dependiendo de la experticia que tenga el programador en la herramienta que seleccione para su desarrollo. Esto debido a que las personas que van a crear los *Plugins* deben tener una experiencia técnica amplia en cuanto a la herramienta que seleccionen para desarrollar los *Plugins*, ya que una persona que no posea dicho conocimiento, requerirá una curva de aprendizaje importante, la cual demanda más tiempo y así afecta su experiencia para la creación de los *Plugins* a través del modelo propuesto. En cuanto a los usuarios que solo reutilizan los *Plugins*, se observó que existe una variación más amplia de perfiles, esto debido a que una persona con conocimientos básicos del entorno de desarrollo puede implementar fácilmente los *Plugins*.
- A partir de la evaluación realizada en la investigación el desarrollo de la presente investigación, fue de gran motivación realizarla debido a que surgieron un sin número de inconvenientes consideradas como lecciones aprendidas. Todos aquellos inconvenientes son siempre necesarios en todo proceso investigativo. Se determinaron a gran escala, grandes inconvenientes en temas como DetectionAR, Tracking de realidad aumentada, los fallos en los tipos de herramientas usados como SDK que se acoplen de la mejor manera a la técnica AR, fallos en cuanto a algunos entornos de desarrollo y su bajo acople con demás tecnologías, entre otros. Todos y cada uno de falencias encontradas permitieron hacer de esta investigación una toma de decisiones que retroalimentan a obtener productos de representación de buena calidad.

#### 6.1.4. REFLEXIÓN CRÍTICA DE LA INVESTIGACIÓN

- Uno de los pilares principales de la ciencia es la observación, actualmente se suele decir que saber observar es saber seleccionar. Y precisamente debido a esto, en la presente investigación, la observación científica permitió la verificación del



comportamiento que tomaban los *Plugins* al momento de ser reutilizarlos y personalizados en diferentes contextos.

- Los conceptos de personalización y reutilización, se evaluaron teniendo en cuenta la observación como método de recolección de información. Esto es debido a que los conceptos pueden ser fácilmente mal interpretados. Para la presente investigación, se tiene en cuenta que la personalización en un *Plugin*, solo compete la variación en el color, textura o tamaño de la representación para realidad aumentada, precisamente esa fue la observación realizada en cada uno de las evaluaciones a los usuarios.
- El concepto de reutilización es un concepto es muy amplio y los usuarios que fueron evaluados podrían mal interpretar su definición. Para la presente investigación se tiene en cuenta que la reutilización es el proceso que realiza la persona evaluada, en el momento en que ellos toman el *Plugin*, lo importan en un proyecto creado en el entorno de desarrollo Unity y lo colocan debidamente en una escena, en ese momento la persona logro la reutilización. Dicho lo anterior, se destaca que la evaluación de las características buscadas en los *Plugins* creados, no se podían dejar a libre interpretación de los evaluados, pues fácilmente podrían generar inconsistencias en los resultados obtenidos.
- Por ultimo cabe destacar que si bien la evaluación de personalización y reutilización arrojaron resultados favorables, se debe en gran parte a la correcta interpretación de los conceptos para medirlos. Aunque quizás, otros investigadores pudiesen considerar que se pudo evaluar aun más a profundidad estos dos indicadores, sin embargo, hay que tener en cuenta que para llevar a cabo esto, se requieren más recursos humanos y económicos, ampliando el alcance establecido de la presente investigación. Por lo tanto, se establecen limitantes para evaluar los indicadores abriendo el paso para establecer trabajos a futuros que se centren en obtener una evaluación más extensa, profunda y concisa de los indicadores mencionados anteriormente.

## 6.2. TRABAJO FUTURO

- Se propone como trabajo futuro la creación de un framework para la WoT sobre el motor de desarrollo Unity3D, considerando que tiene toda la infraestructura necesaria para su desarrollo, de acuerdo a nuestra investigación.
- Se propone generar una segunda interacción de la presente investigación, realizando en el paso de pruebas, listas de chequeos más complejos que a la vez permitan establecer nuevos parámetros de personalización en los *Plugins*, debido que para este proyecto solo se establecen tres (tamaño, color o textura).
- A raíz de un amplio estudio técnico de las herramientas y del proceso de desarrollo de *Plugins* se denoto que todos los *Plugins* implementados necesitaban de una



arquitectura organizacional (ver propuesta de arquitectura en Ilustración 24 ) para sus ficheros internos los cuales contienen toda la lógica de personalización y recursos gráficos. Por lo tanto se propone como trabajo futuro la creación y centralización de una arquitectura genérica de los ficheros contenedores del código fuente de los *Plugins*, pues resultaría ser útil para los desarrolladores de modelos gráficos con base a la organización de sus componentes.

- Se propone generar un tipo de aplicación web, que implemente un buscador ágil de modelos gráficos en el repositorio creado en la presente investigación, de esta manera se permitirá a los desarrolladores, acceder fácilmente a un *Plugin* creado previamente para ser implementado en un proyecto propio. De este modo se hace más óptimo el proceso de reutilización de los *Plugins*, centralizándolos en un repositorio.
- Se considera de vital importancia, tener representaciones graficas de los objetos inteligentes mediante realidad aumentada, con uso de ontologías como perfil de usuario creada por Y.A.Pabón-Guerrero and L.J.Rojas-Bolaños [41] y semánticas que establezcan una interacción entre objetos inteligentes. Estableciendo una representación global de los objetos inteligentes para la WoT, logrando una plena interoperabilidad al combinar las representaciones graficas mediante modelos personalizables y reutilizables pero configurables directamente desde ontologías.
- Al realizar el desarrollo de los *Plugins*, se pudo observar que los *Plugins* requieren de una programación más robusta para impedir cambios en sus componentes por parte de los usuarios que los utilicen, por lo tanto se propone generar una lógica que impida cambios en el tiempo, pero que proporcione los componentes necesarios para una personalización adecuada de la representación.





## REFERENCIAS

- [1] M. Weiser, "The computer for the 21st century," *Scientific American*, 1991.
- [2] N. Bari, G. Mani, and S. Berkovich, "Internet of Things as a Methodological Concept," in *Computing for Geospatial Research and Application (COM.Geo), 2013 Fourth International Conference on*, 2013, pp. 48-55.
- [3] Z. Yu and W. Tie-Ning, "Research on the Visualization of Equipment Support Based on the Technology of Internet of Things," in *Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012 Second International Conference on*, 2012, pp. 1352-1357.
- [4] S. S. Mathew, Y. Atif, Q. Z. Sheng, and Z. Maamar, "Web of Things: Description, Discovery and Integration," in *Internet of Things (iThings/CPSCoM), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, 2011, pp. 9-15.
- [5] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communications Surveys & Tutorials*, 2013.
- [6] J. W. Wu, D. W. Chou, and J. R. Jiang, "The Virtual Environment of Things (VEoT): A Framework for Integrating Smart Things into Networked Virtual Environments," in *Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing (CPSCom), IEEE*, 2014, pp. 456-459.
- [7] L. Mainetti, V. Mighali, and L. Patrono, "A Software Architecture Enabling the Web of Things," *IEEE Internet of Things Journal*, vol. 2, pp. 445-454, 2015.
- [8] IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture," *IEEE Std 802-2001 (Revision of IEEE Std 802-1990)*, pp. 1-48, 2002.
- [9] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 packets over IEEE 802.15.4 networks," 2070-1721, 2007.
- [10] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," 2070-1721, 2014.
- [11] M. A. Niño-Zambrano, "Interacción Semántica de Objetos en la Web de las Cosas," PhD, Departamento de Sistemas, Universidad del Cauca, Popayán, 2016.
- [12] S. Heesuk, H. Seungwook, and L. Dongman, "Contextual Information Provision on Augmented Reality with IoT-Based Semantic Communication," in *Ubiquitous Virtual Reality (ISUVR), 2012 International Symposium on*, 2012, pp. 46-49.
- [13] D. W. F. Van-Krevelen and R. Poelman, *A survey of augmented reality technologies, applications and limitations: kjcomps.6te.net*, 2010.
- [14] T. Leppanen, A. Heikkinen, A. Karhu, E. Harjula, J. Rieki, and T. Koskela, "Augmented Reality Web Applications with Mobile Agents in the Internet of Things," in *2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*, 2014, pp. 54-59.
- [15] M. C. Domingo, "An overview of the Internet of Things for people with disabilities," *Journal of Network and Computer Applications*, vol. 35, pp. 584-596, 2012.
- [16] O. E. Guzmán-Álvarez, "Interfaz de realidad aumentada para la internet de las cosas," Departamento de Sistemas, Universidad del Cauca, 2016.
- [17] D. Riobamba y S. Guerrero, "Escenarios de interacción semántica de objetos inteligentes en la Web de las Cosas," Departamento de Telemática, Universidad del Cauca, 2016.



- [18] S. Electric, "IoT 2020 Business Report. The future of the Internet of Things: From sensor to business sense," 2016.
- [19] M. H. Asghar, A. Negi, and N. Mohammadzadeh, "Principle application and vision in Internet of Things (IoT)," in *Computing, Communication & Automation (ICCCA), 2015 International Conference on*, 2015, pp. 427-431.
- [20] H. Zimmermann, "OSI reference model--The ISO model of architecture for open systems interconnection," *IEEE Transactions on communications*, vol. 28, pp. 425-432, 1980.
- [21] D. Guinard, "Web of Things vs Internet of Things," vol. 1, 2016.
- [22] F. Hu, D. Xie, and S. Shen, "On the application of the internet of things in the field of medical and health care," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, 2013, pp. 2053-2058.
- [23] M. Mrissa, L. Médini, and J.-P. Jamont, "Semantic discovery and invocation of functionalities for the web of things," in *WETICE Conference (WETICE), 2014 IEEE 23rd International*, 2014, pp. 281-286.
- [24] P. E. Estrada-Martinez and J. A. Garcia-Macias, "Semantic Interactions in the Internet of Things," *Int. J. Ad Hoc and Ubiquitous Computing*, vol. 9, p. 1, 2012.
- [25] A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," in *International Symposium on Handheld and Ubiquitous Computing*, 1999, pp. 304-307.
- [26] B. Giullia, "A semantic model for socially aware objects," *Advances in Internet of Things*, vol. 2012, 2012.
- [27] A. A. Migdal, A. Guera-Arcas, and A. Lebedev, "System and method for computer modeling of 3D objects and 2D images by mesh constructions that incorporate non-spatial data such as color or texture," United States Patent, 2001.
- [28] T. Berners-Lee and J. Jaffe, "W3C - World Wide Web," 1994.
- [29] T. Araújo, M. Abayazid, M. J. C. M. Rutten, and S. Misra, "Segmentation and three-dimensional reconstruction of lesions using the automated breast volume scanner (ABVS)," *THE INTERNATIONAL JOURNAL OF MEDICAL ROBOTICS AND COMPUTER ASSISTED SURGERY*, 12 Julio 2016 2016.
- [30] L. Correa-Alfaro, "Comunicabilidad, paradigma de la Interacción Humano-Computador," *No Solo Usabilidad*, 2010.
- [31] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic, "Augmented reality technologies, systems and applications," *Multimedia Tools and ...*, 2011.
- [32] P. Milgram y F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE TRANSACTIONS on Information and Systems*, vol. 77, pp. 1321-1329, 1994.
- [33] R. Azuma, "A survey of augmented reality," *Presence: Teleoperators and virtual environments*, vol. 6, pp. 355-385, 1997.
- [34] D. Guinard, V. Trifa, T. Pham, and O. Liechti, "Towards physical mashups in the Web of Things," in *Networked Sensing Systems (INSS), 2009 Sixth International Conference on*, 2009, pp. 1-4.
- [35] A. Hernandez-Murguía. (2012). *Mashups: Qué Son y Qué No Son*. Available: <https://sg.com.mx/content/view/256/>
- [36] J. Perez and M. Merino. (2015). *Definición de Plugin*. Available: <http://definicion.de/plugin/>
- [37] U. Technologies. (2017). *Prefabs*. Available: <https://docs.unity3d.com/Manual/Prefabs.html>



- [38] G.Fortino, A.Guerrieri, W.Russo, and C.Savaglio, "Integration of agent-based and cloud computing for the smart objects-oriented IoT," in *Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on*, 2014, pp. 493-498.
- [39] A.Martínez-Ospina, "Realidad Aumentada en Interiores: posicionamiento del usuario en dispositivos móviles y aplicaciones en rehabilitación y guiado (GuIAR)," Informatica, 2013.
- [40] M. A. Niño-Zambrano, C. A. Cobos-Lozada, M. A. Ardila-Núñez, W. A. Cháves-Hernández, and L. E. Molina-Romero, "Metamodelo de evaluación para la educación en línea," 2003.
- [41] Y.A.Pabón-Guerrero and L.J.Rojas-Bolaños, "Modelo Semántico para el Manejo de Perfil de Usuario en la IoT," 2017.
- [42] L.M.Echeverry-Tobón and L.E.Delgado-Carmona, "Caso Práctico de la Metodología Ágil XP al Desarrollo de Software," Universidad Tecnológica de Pereira, 2007.
- [43] R. Sampieri Hernández, C. Fernández Collado, and P. Baptista Lucio, "Metodología de la investigación . México DF," ed: México: McGraw-Hill, 2014.